# Finitary non-compositional proof systems for ASL in first-order and higher-order logic

Nikos Mylonakis

March 20, 2000

**Abstract**

In this paper we present finitary proof systems for the deduction of sentences from algebraic specifications inductively defined by specification expresssions in first-order and higher-order logic. Mainly, we redesign the proof systems presented in [4] and in [3] for the reachability and behavioural operators. The main application of the result is to give an adequate representation of this kind of proof systems in a type-theoretic logical framework.

## 1 Introduction

In this paper, we present a finitary version of a certain kind of non-compositional and infinitary proof systems for $ASL$ for the deductions of sentences of a first-order logic from specifications presented in [4] and in [3]. The finitary version of these proof systems can be used to give adequate representations of the proof systems in logical frameworks [7].

In [4] and in [3], the proof systems for the deduction of sentences from specifications are divided in compositional and non-compositial proof systems. Basically, compositional proof systems have the property that the proofs of the deduction of sentences from specification expressions are in terms of the proofs of the deduction of sentences from the subspecifications of the given specification expression. In non-compositional proof systems these kind of proofs are developed with a kind of proof systems which the structure of the proof does not follow in general the structure of the specification expression. Two different kind of non-compositional proof systems have been developed: one of them are defined in terms of the normal form of the given specification expression and the others are inductively defined by specification expression adding specific rules and axioms for each case.

In this paper, we concentrate on non-compositional proof systems inductively defined for specification expressions for $ASL$ in first-order and higher-order logic. The main differences between the original infinitary proof systems of $ASL$ and the finitary ones presented here are in the behavioural operators.

Different kind of non-compositional proof systems inductively defined by specification expressions have been developed for the behavioural operators. In [5], these operators can just be applied to basic specifications, whereas in [4] they can be applied to any kind of specification expressions. In [5] it is developed specific proof systems for the behaviour and the abstract operator for standard and behavioural theories and they use higher-order logic as specification logic, whereas in [4] they present proof systems for the behaviour, abstract and quotient operator but just for standard theories and they use finitary and infinitary first-order logic. In this paper, we will present in the same way in first-order and higher-order logic, the proof systems for basic specifications and for the sum, rename and export operator, and after that we will present two different proof systems for the rest of the operators: one with the institution [2] $FOLEQ$ and the other with the institution $HOL$. As we have mentioned , the design of the proof systems is based on [3] and the axiomatisation of the observational equality associated to the behavioural operators for the proof system of higher-order logic is based on [5].

Basically, the proof systems for the behavioural operators are defined as the proof systems of structured specifications whose semantics are equivalent to the semantics of the behavioural operators. The result which states the equivalence is presented in the paper for the proof system of higher-order logic. For first-order logic, the proof system is designed in a similar way but instead of having an axiomatisation of the observational equality in first-order logic, we have proof rules to derive proofs about observational equality.

The structure of the paper is as follows: first we present the two institutions of first-order and higher-order logic, then we present the algebraic specification language $ASL$ together with the equalities associated to the behavioural operators, next we present the proof systems for first-order and higher-order logic and finally we raise some conclusions.

# 2 The first-order and higher-order institutions

In this section we present the institutions $FOLEQ$ and $HOL$ which will be used in the rest of the paper:

**Proposition 2.1** *The tuple*

$$FOLEQ = (AlgSig, Sen_{FOLEQ}, Alg, <\models_{FOLEQ,\Sigma}>_{\Sigma \in |AlgSig|})$$

*such that:*

- *The category of first-order relational signatures $AlgSig$ whose objects are first-order relational signatures and morphisms are signature morphisms.*
- $Sen_{FOLEQ} : AlgSig \rightarrow Set$ *is a functor defined in the following way:*
  - *For each $\Sigma = (S, Op) \in |AlgSig|$, the set $Sen(\Sigma)$ is inductively defined by the following set of rules:*

* $\mathbf{true}, \mathbf{false} \in Sen(\Sigma)$.

* *If* $t, r \in T_\Sigma(X)_s$ *for* $s \in Sorts(\Sigma)$ *then* $t = r \in Sen(\Sigma)$.

* *If* $p : s_1 \times \ldots \times s_n \in Prs(\Sigma)$ *and* $t_1 \in T_\Sigma(X)_{s_1}, \ldots, t_n \in T_\Sigma(X)_{s_n}$ *then* $p(t_1, \ldots, t_n) \in Sen(\Sigma)$

* *If* $\phi, \psi \in Sen(\Sigma)$ *then* $\neg\, \phi$, $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \supset \psi \in Sen(\Sigma)$.

* *If* $x \in X_s$ *and* $\phi \in Sen(\Sigma)$ *then* $\forall x : s.\phi, \exists x : s.\phi \in Sen(\Sigma)$

– *For each signature morphism* $\sigma : \Sigma \to \Sigma'$ *the morphism*

$$Sen_{FOLEQ}(\sigma) : Sen_{FOLEQ}(\Sigma) \to Sen_{FOLEQ}(\Sigma')$$

*is the usual renaming function between first-order sentences.*

- *the functor* $Alg : AlgSig^{op} \to Cat$ *where:*
  - *for any* $\Sigma \in |AlgSig|$, $Alg(\Sigma)$ *is the category of* $\Sigma$*-algebras*
  - *for any morphism* $\sigma : \Sigma \to \Sigma'$ *in* $AlgSig$, $Alg(\sigma)$ *is the reduct functor* $\_|_\sigma : Alg(\Sigma') \to Alg(\Sigma)$.
- *for each* $\Sigma \in |AlgSig|$ *the satisfaction relation* $\models_{FOLEQ,\Sigma}$ *is defined as follows:*

$$\forall A \in |Alg(\Sigma)|.\forall \phi \in Sen_{FOLEQ}(\Sigma).A \models_{FOLEQ,\Sigma} \phi \Leftrightarrow$$

$$\forall \rho \in X \to A.A \models_{FOLEQ,\Sigma,\rho} \phi$$

*where for any* $\Sigma \in |AlgSig|$ *and for any* $\rho : X \to A$, *the relation* $\models_{FOLEQ,\Sigma,\rho}$ *is simultaneously defined by induction as follows:*

– $A \models_\rho \mathbf{true}$ *holds.*

– *If* $I_\rho(t) = I_\rho(r)$ *then* $A \models_\rho t = r$ *holds.*

– *If* $A \models_\rho \phi$ *does not hold then* $A \models_\rho \neg\, \phi$ *holds.*

– *If* $A \models_\rho \phi$ *holds and* $A \models_\rho \psi$ *holds then* $A \models_\rho \phi \wedge \psi$ *holds.*

– *If* $A \models_\rho \phi$ *holds or* $A \models_\rho \psi$ *holds then* $A \models_\rho \phi \vee \psi$ *holds.*

– *If* $A \models_\rho \phi$ *does not hold or* $A \models_\rho \psi$ *holds then* $A \models_\rho \phi \supset \psi$ *holds.*

– *If* $\forall v \in A_s . A \models_{\rho \cup \{(x,v)\}} \phi$ *holds then* $A \models_\rho \forall x : s.\phi$ *holds.*

– *If* $A \models_{\rho \cup \{(x,v)\}} \phi$ *holds for some* $v \in A_s$ *then* $A \models_\rho \exists x : s.\phi$ *holds.*

*is an institution.*

Before presenting an algebraic institution for higher-order logic ($HOL$), we give some basic definitions which will be used in its definition.

**Definition 2.1** *For each* $\Sigma = (S, Op, Pr) \in |AlgSig|$, *the set* $Types_{HOL}(\Sigma)$ *is inductively defined by the following set of rules:*

- *If* $s \in S$ *then* $s \in Types_{HOL}(\Sigma)$.

- *If* $\tau_1 \in Types_{HOL}(\Sigma), \ldots, \tau_n \in Types_{HOL}(\Sigma)$ *and* $n \geq 0$ *then* $[\tau_1, \ldots, \tau_n] \in Types_{HOL}(\Sigma)$.

**Notation:** *The type* [] *will be normally denoted by* **Prop**.

*For any signature morphism* $\sigma : \Sigma \to \Sigma'$, *we will also denote by* $\sigma$ *the usual renaming function between types* $\sigma : Types_{HOL}(\Sigma) \to Types_{HOL}(\Sigma')$.

**Definition 2.2** *The semantic function* $[\![\tau]\!]_A$ *is inductively defined for any type* $\tau \in Types_{HOL}(\Sigma)$ *and for any* $\Sigma$*-algebra* $A$ *as follows:*

$$[\![s]\!]_A = A_s$$

$$[\![[\tau_1, \ldots, \tau_n]]\!]_A = \mathcal{P}([\![\tau_1]\!]_A \times \ldots \times [\![\tau_n]\!]_A)$$

**Notation:** *The semantics of* **Prop** *is a set of two elements: the empty set and the set with the empty tuple. These two elements will be denoted as* **ff** *and* **tt** *respectively.*

**Definition 2.3** *The set* $Sen_{HOL}(\Sigma, X_{HOL}, \tau)$ *for a given* $Types_{HOL}$ $(\Sigma)$*-sorted infinite denumerable set of variables* $X_{HOL}$ *and for every* $\tau \in Types_{HOL}(\Sigma)$ *is inductively defined by the following set of rules:*

- *If* $x \in X_{HOL,\tau}$ *then* $x_\tau \in Sen_{HOL}(\Sigma, X_{HOL}, \tau)$.

- *If* $f : s_1 \times \ldots \times s_n \to s \in Ops(\Sigma)$, $t_1 \in T_{\Sigma,s_1}(< X_{HOL,s} >_{s \in S}), \ldots,$

  $t_n \in T_{\Sigma,s_n}(< X_{HOL,s} >_{s \in S})$

  *then* $f(t_1, \ldots, t_n) \in Sen_{HOL}(\Sigma, X_{HOL}, s)$.

- *If* $p : s_1 \times \ldots \times s_n \in Prs(\Sigma)$, $t_1 \in T_{\Sigma,s_1}(< X_{HOL,s} >_{s \in S}), \ldots,$

  $t_n \in T_{\Sigma,s_n}(< X_{HOL,s} >_{s \in S})$

  *then* $p(t_1, \ldots, t_n) \in Sen_{HOL}(\Sigma, X_{HOL}, \mathbf{Prop})$

- *If* $\tau_1, \ldots, \tau_n \in Types_{HOL}(\Sigma)$, $x_1 \in X_{HOL,\tau_1}, \ldots, x_n \in X_{HOL,\tau_n}$

  *and* $\phi \in Sen_{HOL}(\Sigma, X_{HOL}, \mathbf{Prop})$ *then*

  $\lambda(x_1 : \tau_1, \ldots, x_n : \tau_n).\phi \in Sen_{HOL}(\Sigma, X_{HOL}, [\tau_1, \ldots, \tau_n])$.

- *if* $\tau_1, \ldots, \tau_n \in Types_{HOL}(\Sigma)$, $t \in Sen_{HOL}(\Sigma, X_{HOL}, [\tau_1, \ldots, \tau_n])$,

  $t_1 \in Sen_{HOL}(\Sigma, X_{HOL}, \tau_1), \ldots, t_n \in Sen_{HOL}(\Sigma, X_{HOL}, \tau_n)$

  *then* $t(t_1, \ldots, t_n) \in Sen_{HOL}(\Sigma, X_{HOL}, \mathbf{Prop})$.

- *if* $\tau \in Types_{HOL}(\Sigma)$, $x \in X_{HOL,\tau}$ *and*

  $\phi \in Sen_{HOL}(\Sigma, X_{HOL}, \mathbf{Prop})$ *then*

  $\forall x : \tau.\phi \in Sen_{HOL}(\Sigma, X_{HOL}, \mathbf{Prop})$.

- *if* $\phi, \phi' \in Sen_{HOL}(\Sigma, X, \mathbf{Prop})$ *then* $\phi \supset \phi' \in Sen_{HOL}(\Sigma, X, \mathbf{Prop})$.

**Notation:** *We will denote by* $Terms_{HOL}(\Sigma, X_{HOL})$ *the set*

$$\bigcup_{\tau \in Types_{HOL}(\Sigma)} Sen_{HOL}(\Sigma, X_{HOL}, \tau)$$

*For any signature morphism* $\sigma : \Sigma \to \Sigma'$, *we will also denote by* $\sigma$ *the usual renaming function between terms*

$$\sigma : Terms_{HOL}(\Sigma, X_{HOL}) \to Terms_{HOL}(\Sigma', X_{HOL})$$

*such that for any type* $\tau \in Types_{HOL}(\Sigma)$, *for any higher-order sentence* $\phi \in Sen_{HOL}(\Sigma, X_{HOL}, \tau)$, $\sigma(\phi) \in Sen_{HOL}(\Sigma', X_{HOL}, \sigma(\tau))$.

*The usual definition of* $\beta-$*equality between terms in* $Terms_{HOL}(\Sigma, X_{HOL})$ *identifying also* $\alpha$*-convertible terms will be denoted by* $=_\beta$ *and the usual substitution operation avoiding name clashes will be denoted by* $t\{t'/x\}$ *for any* $t \in Terms_{HOL}(\Sigma, X_{HOL})$, $t' \in Sen_{HOL}(\Sigma, X_{HOL}, \tau)$ *and* $x \in X_{HOL,\tau}$.

**Definition 2.4** *The function* $[\![t]\!]_{\rho,A}$ *for any term* $t \in Terms_{HOL}(\Sigma, X_{HOL})$, *for any algebra* $A \in Alg(\Sigma)$, *for any* $Types_{HOL}(\Sigma)$*-sorted valuation* $\rho$ *which for every* $\tau \in Types_{HOL}(\Sigma)$, $\rho_\tau$ *has arity* $\rho_\tau : X_{HOL,\tau} \to [\![\tau]\!]_A$ *is inductively defined by the structure of* $t$ *as follows:*

$[\![x_\tau]\!]_{\rho,A} = \rho_\tau(x)$

$[\![f(t_1, \ldots, t_n)]\!]_{\rho,A} = f_A([\![t_1]\!]_{\rho,A}, \ldots, [\![t_n]\!]_{\rho,A})$

$[\![p(t_1, \ldots, t_n)]\!]_{\rho,A} = if\ ([\![t_1]\!]_{\rho,A}, \ldots, [\![t_n]\!]_{\rho,A}) \in p_A\ then\ \mathbf{tt}\ else\ \mathbf{ff}$

$[\![\lambda(x_1 : \tau_1, \ldots, x_n : \tau_n).\phi]\!]_{\rho,A} =$

$\{(v_1, \ldots, v_n) | v_1 \in [\![\tau_1]\!]_{\rho,A}, \ldots, v_n \in [\![\tau_n]\!]_{\rho,A}, [\![\phi]\!]_{\rho \cup \{(x_1,v_1),\ldots,(x_n,v_n)\}} = \mathbf{tt}\}$

$[\![t\ (t_1, \ldots, t_n)]\!]_{\rho,A} =$

$if\ ([\![t_1]\!]_{\rho,A}, \ldots, [\![t_n]\!]_{\rho,A}) \in [\![t]\!]_{\rho,A}\ then\ tt\ else\ ff$

$[\![\phi \supset \phi']\!]_{\rho,A} = if\ [\![\phi]\!]_{\rho,A} = \mathbf{tt}\ then\ [\![\phi]\!]_{\rho,A}\ else\ \mathbf{tt}$

$[\![\forall x : \tau.\phi]\!]_{\rho,A} = if\ \forall v \in [\![\tau]\!]_A.[\![\phi]\!]_{\rho \cup \{(x,v)\},A} = \mathbf{tt}\ then\ \mathbf{tt}\ else\ \mathbf{ff}$

**Proposition 2.2** *The tuple*

$$HOL = (AlgSig, Sen_{HOL}, Alg, <\models_{HOL,\Sigma}>_{\Sigma \in |AlgSig|})$$

*such that:*

- *The category of first-order relational signatures* $AlgSig$ *whose objects are first-order relational signatures and morphisms are signature morphisms.*
- $Sen_{HOL} : AlgSig \to Set$ *is a functor defined in the following way:*
  - *For each* $\Sigma \in |AlgSig|$, *the set* $Sen_{HOL}(\Sigma)$ *is defined as*

    $$Sen_{HOL}(\Sigma) = Sen_{HOL}(\Sigma, X_{HOL}, \mathbf{Prop})$$

    *for a given* $Types_{HOL}$ $(\Sigma)$*-sorted infinite denumerable set of variables* $X_{HOL}$
  - *For each signature morphism* $\sigma : \Sigma \to \Sigma'$ *the morphism*

    $$Sen_{HOL}(\sigma) : Sen_{HOL}(\Sigma) \to Sen_{HOL}(\Sigma')$$

    *is the usual renaming function between sentences using* $\sigma : \Sigma \to \Sigma'$ *and it will also be denoted just by* $\sigma$.
- *the functor* $Alg : AlgSig^{op} \to Cat$ *where:*

- *for any* $\Sigma \in |AlgSig|$, $Alg(\Sigma)$ *is the category of* $\Sigma$*-algebras*
- *for any morphism* $\sigma : \Sigma \to \Sigma'$ *in* $AlgSig$, $Alg(\sigma)$ *is the reduct functor* $\_|_\sigma : Alg(\Sigma') \to Alg(\Sigma)$.

- *for each* $\Sigma \in |AlgSig|$, *for all* $A \in |Alg(\Sigma)|$, *for all* $\phi \in Sen_{HOL}(\Sigma)$, *the satisfaction relation* $A \models_\Sigma \phi$ *holds if and only if for any* $Types_{HOL}(\Sigma)$*-sorted valuation* $\rho$ *which for every* $\tau \in Types_{HOL}(\Sigma)$, $\rho_\tau$ *has arity* $X_{HOL,\tau} \to [\![\tau]\!]_A$, $[\![\phi]\!]_{\rho,A} = \mathbf{tt}$

*is an institution.*

# 3 ASL

In this section we present the semantics of $ASL$ similar to [3]. Since the semantics is almost the same for $FOLEQ$ and for $HOL$, we present both in an uniform way. In the definition, $INSFH$ ranges over $FOLEQ$ and $HOL$.

**Definition 3.1** *The syntax of the operators of ASL is the following:*

$$
\begin{aligned}
SP_0 ::= \quad & < \Sigma, \Phi > \\
& SP_1|_\Sigma \\
& SP_1 +_\Sigma SP_2 \\
& \mathit{rename} \quad SP \quad \mathbf{by} \quad \sigma \\
& \mathbf{reach} \quad SP \quad \mathbf{with} \quad (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R}) \\
& \mathbf{behaviour} \quad SP \quad \mathbf{wrt} \quad \approx_{Obs,In} \\
& \mathbf{abstract} \quad SP \quad \mathbf{by} \quad \equiv_{Obs,In} \\
& SP/\approx_{Obs,In}
\end{aligned}
$$

*where the signature* $\Sigma = (S, Op) \in |AlgSig|$, $\Phi \subseteq Sen_{INSFH}(\Sigma)$ , $\sigma$ *is a signature morphism,* $\approx_{Obs,In}$ *is the observational equality formally defined below,* $\equiv_{Obs,In}$ *is the behavioural equality formally defined below and Obs and In are a set of sorts.. The semantics of the ASL operators is inductively defined*

*as follows:*

$Signature(< \Sigma, \Phi >) = \Sigma$

$Models(< \Sigma, \Phi >) = \{A \mid A \models_{FOL,\Sigma} \Phi\}$

*where the signature* $\Sigma = (S, Op) \in |AlgSig|$ *and* $\Phi \subseteq |Sen_{INSFH}(\Sigma)|$.

$Signature(rename \quad SP \quad \mathbf{by} \quad \sigma) = \Sigma$

$Models(rename \quad SP \quad \mathbf{by} \quad \sigma) = \{ A \in Alg(\Sigma) \mid A|_\sigma \in Models(SP) \}$

*where* $\sigma : Signature(SP) \rightarrow \Sigma$ *is a bijective signature morphism.*

$Signature(SP|_\Sigma) = \Sigma$

$Models(SP|_\Sigma) = \{A|_\Sigma \mid A \in Models(SP)\}$

*where the signature* $\Sigma = (S, Op) \in |AlgSig|$ *and* $\Sigma \subseteq Signature(SP)$

$Signature(SP_1 +_\Sigma SP_2) = Signature(SP_1) +_\Sigma Signature(SP_2)$

$Models(SP_1 +_\Sigma SP_2) =$

$\{A \mid A \in Alg(Signature(SP_1) +_\Sigma Signature(SP_2)),$

$A|_{inl} \in Models(SP_1), A|_{inr} \in Models(SP_2)\}$

*where the signature* $\Sigma = (S, Op) \in |AlgSig|$, $SP_1, SP_2$ *ranges over specification expressions in* $SPEX(ASLK)$, $\Sigma \subseteq Signature(SP_1)$, $\Sigma \subseteq Signature(SP_2)$ *and* $Signature(SP_1) +_\Sigma Signature(SP_2)$ *is the pushout of the two obvious inclusions between* $\Sigma$ *and* $Signature(SP_1)$ *and* $\Sigma$ *and* $Signature(SP_2)$

$Signature(\textbf{reach} \quad SP \quad \textbf{with} \quad (\mathcal{S}_{\mathcal{R}}, \mathcal{F}_{\mathcal{R}})) = Signature(SP)$

$Models(\textbf{reach} \quad SP \quad \textbf{with} \quad (\mathcal{S}_{\mathcal{R}}, \mathcal{F}_{\mathcal{R}})) = \{A \in Models(SP) \mid A \models (\mathcal{S}_{\mathcal{R}}, \mathcal{F}_{\mathcal{R}})\}$

$Signature(\textbf{behaviour} \quad SP \quad \textbf{wrt} \quad \approx_{Obs,In}) = Signature(SP)$

$Models(\textbf{behaviour} \quad SP \quad \textbf{wrt} \quad \approx_{Obs,In}) = \{ \ A/\approx_{Obs,In} \ \mid A \in Models(SP) \ \}$

$where \ In, Obs \subseteq Sorts(Signature(SP))$

$Signature(\textbf{abstract} \quad SP \quad \textbf{by} \quad \equiv_{Obs,In}) = Signature(SP)$

$Models(\textbf{abstract} \quad SP \quad \textbf{by} \quad \equiv_{Obs,In}) = \{A \mid \exists B \in Models(SP).B \equiv_{Obs,In} A\}$

$where \ In, Obs \subseteq Sorts(Signature(SP))$

$Signature(SP/\approx_{Obs,In}) = Signature(SP)$

$Models(SP/\approx_{Obs,In}) = \{A \mid \exists B \in Models(SP)/\approx_{Obs,In} .B \cong A\}$

$where \ In, Obs \subseteq Sorts(Signature(SP))$

*and the formal definitions of the observational equality and the behavioural equality between algebras are the following:*

**Definition 3.2** *Given a signature $\Sigma = (S, Op) \in |AlgSig|$ and a set of sorts $In \subseteq Sorts(\Sigma)$, $\Sigma$ is sensible wrt $In$ if for all $s \in Sorts(\Sigma) - In$ , there exists a term $t$ of sort $s$ built with function symbols in $\Sigma$ and variables of sort $s \in In$.*

**Definition 3.3** *Let $\Sigma = (S, Op)$ be a signature in $|AlgSig|$, let $In$ and $Obs$ be two set of sorts s.t. $In, Obs \subseteq Sorts(\Sigma)$ and let $X_{In}$ be an $In$-sorted set of variables. The $Sorts(\Sigma)$-sorted set of contexts $C_{\Sigma,Obs}(X_{In})$ is defined for each sort $s$ as the set of terms $T_{\Sigma}(X_{In} \cup z_s)$ of result sort in $Obs$ such that $z_s$ is a free variable which satisfies the condition $\{z_s\} \cap X_s = \emptyset$. This set is also denoted as $C_{\Sigma,Obs}(X_{In}, z_s)$ for every sort $s \in S$.*

**Definition 3.4** *Let $\Sigma = (S, Op)$ be a signature in $|AlgSig|$, let $Obs$ and $In$ be two set of sorts s.t. $Obs, In \subseteq Sorts(\Sigma)$ and $In$ is sensible wrt $\Sigma$. Let $A$ be a $\Sigma$-algebra. The observational equality ($\approx_A^{Obs,In}$) is formally defined for each*

*sort $s$ and for each $v, w \in A[X_{In}]_s$ as follows:*

$$v \approx_{s,A}^{Obs,In} w \Leftrightarrow$$

$$\begin{cases} \forall c \in C_{\Sigma,Obs}(X_{In}, z_s).\forall \alpha \in X_{In} \to A[X_{In}]. \\ \qquad I_{\alpha \cup \{(z_s,v)\}}(c) = I_{\alpha \cup \{(z_s,w)\}}(c) \quad , if s \in S - Obs \\ \\ v = w \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad , if s \in Obs \end{cases}$$

**Proposition 3.1** *Let $\Sigma = (S, Op)$ be a signature in $|AlgSig|$, let $Obs$ and $In$ be two set of sorts s.t. $Obs, In \subseteq Sorts(\Sigma)$. The observational equality ($\approx^{Obs,In}$) is a family of partial $\Sigma$-congruences.*

**Definition 3.5** *Let $\Sigma = (S, Op)$ be a relational signature in $|AlgSig|$, let $In$ and $Obs$ be two sets of sorts s.t. $In, Obs \subseteq Sorts(\Sigma)$ and let $X_{In}$ be an $In$-sorted set of variables. The behavioural equality between $\Sigma$-algebras $\equiv_{Obs,In}$ is formally defined as:*

$$A \equiv_{Obs,In} B \quad \Leftrightarrow \quad \forall t, r \in T_\Sigma(X_{In}).\forall \alpha \in X_{In} \to A.\forall \beta \in X_{In} \to B.$$

$$I_\alpha(t) = I_\alpha(r) \Leftrightarrow I_\beta(t) = I_\beta(r)$$

**Proposition 3.2** *Let $\Sigma = (S, Op)$ be a signature in $|AlgSig|$, and let $In$ and $Obs$ be two sets of sorts s.t. $In, Obs \subseteq Sorts(\Sigma)$.$\equiv_{Obs,In}$ is an equivalence relation.*

# 4 Proof systems inductively defined by specification expressions

In this section, we present this kind of proof systems for $ASL$ with the institutions $FOLEQ$ and $HOL$. We present first some basic notions of natural deduction systems and consequence relations and then we present the non-compositional proof systems. First we present the proof systems for the operators which have the same formulation for $FOLEQ$ and $HOL$ and after that we present the proof systems for the rest of the operators first for $FOLEQ$ and then for $HOL$.

Our proof systems will be formulated as natural deduction systems See [1] for a formal description of these systems. Basically, these systems are defined by a finite set of natural deduction rules. These kind of rules are defined by a set of n premises, a conclusion, and side conditions are allowed. *Premises* and *conclusions* are defined by sequents with schematic variables and therefore a rule denotes in general a set of (n+1)-tuples of sequents. An *instance* of a rule

is a (n+1)-tuple of sequents of this set. In general, sequents are defined with judgements.

The judgements which we will use for the definition of the proof systems for first-order and higher-order logic are:

- $\phi$. This judgement means that *the formula $\phi$ holds.*
- $\phi : \tau$. This judgement will be used for the definition of $\Pi_{HOL}$ and it means that *the higher-order formula $\phi$ has type $\tau$.*
- $\phi =_{\beta, X_{FOLEQ}} \phi$. This judgement will be used for the definition of $\Pi_{HOL}$ and it means that *the higher-order formulas $\phi$ and $\phi'$ are equivalent by $\beta$-equality.*

The proof system for first-order logic ($\Pi_{FOLEQ}(\Gamma, \Sigma)$) for any set of sentences $\Gamma$ with signature $\Sigma$ will be formulated as a natural deduction system with the only sequent $\Gamma \Rightarrow_{X_{FOLEQ}} \phi$ where in this case $X_{FOLEQ}$ is a finite set of pairs of variable and its associated sort and the proof system for higher-order logic ($\Pi_{HOL}(\Gamma, \Sigma)$) for any set of assumptions $\Gamma$ with signature $\Sigma$ will be formulated with the following sequents:

- $\Gamma \Rightarrow_{X_{HOL}} \phi$ where in this case $X_{HOL}$ is is a finite set of pairs of variable and its associated higher-order type
- $X_{HOL} \blacktriangleright \phi : \tau$ where $\phi$ is a higher-order formula and $\tau$ its associated type in $Types_{HOL}(\Sigma)$.
- $\phi =_{\beta, X_{HOL}} \phi'$ where $\phi$ and $\phi'$ are higher-order formulas.

In the following definitions, $INSFH$ will range over $FOLEQ$ and $HOL$:

**Definition 4.1** *The consequence relation* $\vdash_{\Sigma, \Pi_{INSFH}(\Gamma,\Sigma)}$: $\mathcal{P}(|Sen_{INSFH}(\Sigma)|) \times |Sen_{INSFH}(\Sigma)|$ *for a given* $\Sigma \in |AlgSig|$ *is defined as follows:*

$$\Gamma \vdash_{\Sigma} \phi \Leftrightarrow \textit{Any closed sequent of the form } \Gamma \Rightarrow_X \phi$$

$$\textit{has a derivation in } \Pi_{INSFH}(\Gamma, \Sigma).$$

**Definition 4.2** *A consequence relation* $\vdash_{\Sigma, \Pi_{INSFH}(\Gamma,\Sigma)}$: $\mathcal{P}(|Sen_{INSFH}(\Sigma)|) \times |Sen_{INSFH}(\Sigma)|$ *is sound if it satisfies the following condition:*

$$\forall \phi \in Sen_{INSFH}(\Sigma).\Gamma \vdash_{\Sigma, \Pi_{INSFH}(\Gamma,\Sigma)} \phi \Rightarrow$$

$$(\bigwedge_{\psi \in \Gamma} A \models_{\Sigma, INSFH} \psi) \Rightarrow (A \models_{\Sigma, INSFH} \phi)$$

**Definition 4.3** *A consequence relation* $\vdash_{\Sigma, \Pi_{INSFH}(\Gamma,\Sigma)}$: $\mathcal{P}(|Sen_{INSFH}(\Sigma)|) \times |Sen_{INSFH}(\Sigma)|$ *is sound and complete if it satisfies the following condition:*

$$\forall \phi \in Sen_{INSFH}(\Sigma).\Gamma \vdash_{\Sigma, \Pi_{INSFH}(\Gamma,\Sigma)} \phi \Leftrightarrow$$

$$(\bigwedge_{\psi \in \Gamma} A \models_{\Sigma, INSFH} \psi) \Rightarrow (A \models_{\Sigma, INSFH} \phi)$$

Our usual denotation of the proof systems inductively defined by specification expressions will be $\Pi^{ASL}_{INSFH}$ where $INSFH$ is the first-order or higher-order institution. We will denote by $\Pi^{ASL}_{AINS}(SP)$ the proof system associated to the specification expression $SP$ of $ASL$.

Similarly to the consequence relations associated to the proof systems for $FOLEQ$ and $HOL$, we can define consequence relations of sentences of $FOLEQ$ and $HOL$ from specification expressions using non-compositional proof systems. The definition of these consequence relation will use a function $Symbols$, which given a proof system $\Pi^{ASL}_{INSFH}$ (where again $INSFH$ ranges over $FOLEQ$ and $HOL$) and a specification expression $SP \in SPEX(ASL)$ will return the symbols of the proof system $\Pi^{ASL}_{INSFH}(SP)$ since we won't assume that the symbols of the specification expression $SP$ is equal to the symbols of the proof system $\Pi^{ASL}_{INSFH}(SP)$. We will just assume that there exists an inclusion morphism between $Signature(SP)$ and $Symbols(\Pi^{ASL}_{INSFH}, SP)$. Also, the definition of the consequence relation will use the function $\Gamma env$ which given a proof system and an specification expression $SP \in SPEX(ASL)$ will return the set of assumptions which we can use for the derivation of sentences from the given specification expression. Finally, it will also be used the function $Rules$ which given a proof system and an specification expression, it will return the proof rules of the proof system.

For any bijective signature morphism $\sigma : Symbols(\Pi^{ASL}_{INSFH}(SP)) \rightarrow \Sigma'$, we will also denote by $\sigma$ the renaming function which given a proof system $\Pi^{ASL}_{INSFH}(SP)$ will return the resulting proof system of applying to every sentence of every rule of the given proof system the morphism between sentences $\sigma : Sen_{INSFH}\ (Symbols(\Pi^{ASL}_{INSFH}(SP))) \rightarrow Sen_{INSFH}(\Sigma')$.

The definition of the consequence relation associated to these non-compositional proof systems and some of its properties are as follows:

**Definition 4.4** *The consequence relation $\vdash_{\Pi^{ASL}_{INSFH}}$ which relates specification expressions $SP \in SPEX(ASL)$ with sentences $\phi \in Sen(Symbols(\Pi^{ASL}_{INSFH}, SP))$ is defined as follows:*

$$SP \vdash_{\Pi^{ASL}_{INSFH}} \phi \Leftrightarrow \textit{Any closed sequent of the form}$$

$$\Gamma env(\Pi^{ASL}_{INSFH}, SP) \Rightarrow_{Symbols(\Pi^{ASL}_{INSFH}, SP), X} \phi \textit{ has a derivation}$$

$$\textit{in } Rules(\Pi^{ASL}_{INSFH}, SP).$$

**Definition 4.5** *For any signature $\Sigma \in AlgSig$, a $\Sigma$-algebra $A \in Alg(\Sigma)$ satisfies a closed sequent of the form $\Gamma \Rightarrow_X \phi$ if the following condition holds:*

$$(\bigwedge_{\psi \in \Gamma} A \models_{\Sigma, INSFH} \psi) \Rightarrow A \models_{\Sigma, INSFH} \phi$$

**Definition 4.6** *For any signature $\Sigma \in AlgSig$, a $\Sigma$-algebra $A \in Alg(\Sigma)$ satisfies a closed sequent of the form $X \blacktriangleright \phi : \tau$ if $[\![\phi]\!]_{\rho,A} \in [\![\tau]\!]_A$ for any $Types_{HOL}(\Sigma)$-sorted valuation $\rho$ such that for any $x : \tau \in X$, $x \in Dom_\tau(\rho_\tau)$.*

**Definition 4.7** *For any signature $\Sigma \in AlgSig$, a $\Sigma$-algebra $A \in Alg(\Sigma)$ satisfies a closed sequent of the form $\phi =_{X,\beta} \phi'$ if $[\![\phi]\!]_{\rho,A} = [\![\phi']\!]_{\rho,A}$ for any $Types_{HOL}(\Sigma)$-sorted valuation $\rho$ such that for any $x : \tau \in X$, $x \in Dom_\tau(\rho_\tau)$.*

**Definition 4.8** *For any signature $\Sigma \in AlgSig$, a $\Sigma$-algebra $A \in Alg(\Sigma)$ satisfies $\Pi^{ASL}_{INSFH}$ if for all the rules of the natural deduction system the following condition holds:*

- *If $A$ satisfies all the sequents of the premises then $A$ satisfies the sequent of the conclusion.*

**Definition 4.9** *A consequence relation $\vdash_{\Pi^{ASL}_{INSFH}(SP)}$ is sound if for all specification expressions $SP \in SPEX(ASL)$ and for all $A \in Models(SP)$ there exists a $Symbols(\Pi^{ASL}_{INSFH}(SP))$-algebra $A'$ which satisfies the following conditions (which we will refer as soundness conditions):*

- $A'|_{Signature(SP)} = A$.
- $\forall \psi \in \Gamma env(\Pi^{ASL}_{INSFH}, SP).A' \models \psi$.
- *$A'$ satisfies $Rules(\Pi^{ASL}_{INSFH}, SP)$.*

**Proposition 4.1** *If $\phi \in Sen_{AINS}(Signature(SP))$ and $\vdash_{\Pi^{ASL}_{INSFH}(SP)}$ is sound then $SP \vdash_{\Pi^{ASL}_{INSFH}(SP)} \phi \supset SP \models \phi$.*

**Proof:**

Assume that $SP \vdash_{\Pi^{ASL}_{INSFH}(SP)} \phi$.

We have to show that for all $A \in Models(SP)$ $A \models \phi$.

Let A' be the $Symbols(\Pi^{ASL}_{INSFH}, SP)$-algebra which satisfies the soundness conditions for a given $Signature(SP)-$algebra A.

Since $SP \vdash_{\Pi^{ASL}_{INSFH}(SP)} \phi$, there exists a derivation of the sequent

$$\Gamma env(\Pi^{ASL}_{INSFH}, SP) \Rightarrow_{Symbols(\Pi^{ASL}_{INSFH}, SP), X} \phi$$

and since $A'$ satisfies $\Pi^{ASL}_{INSFH}(SP)$ we can deduce that

$$(\bigwedge_{\psi \in \Gamma env(\Pi^{ASL}_{INSFH}, SP)} A' \models_{Symbols(\Pi^{ASL}_{INSFH}, SP)} \psi) \Rightarrow A' \models_{Symbols(\Pi^{ASL}_{INSFH}, SP)} \phi$$

By the second soundness condition, we have that $A' \models_{Symbols(\Pi^{ASL}_{INSFH}, SP)} \phi$ and since $A'|_{Signature(SP)} = A$ and $\phi \in Sen_{INSFH}(Signature(SP))$, we have that $A \models_{Signature(SP)} \phi$.

**Definition 4.10** *A consequence relation $\vdash_{\Pi^{ASL}_{INSFH}(SP)}$ is sound and complete if for all specification expressions $SP \in SPEX(ASL)$, for all sentences $\phi \in |Sen_{INSFH}(Symbols(\Pi^{ASL}_{INSFH}, SP))|$ for all $A \in Models(SP)$, if $A \models \phi$ there exists a $Symbols(\Pi^{ASL}_{INSFH}(SP))$-algebra $A'$ such that $A'$ satisfies the soundness conditions and $A' \models \phi$ if and only if $SP \vdash_{\Pi^{ASL}_{INSFH}(SP)} \phi$.*

## 4.1 Proof system for the operators with an uniform presentation in $FOLEQ$ and in $HOL$

In this subsection, we present the proof systems $\Pi^{ASL}_{FOLEQ}$ and $\Pi^{ASL}_{HOL}$ for basic specifications, the sum, the rename and the export operator.

**Definition 4.11** *The proof system $\Pi_{INSFH}(ASL)$ where $INSFH$ ranges over $FOLEQ$ and $HOL$ is inductively defined for basic specifications, the sum, the export and the rename operator as follows:*

$$Rules(\Pi^{ASL}_{INSFH}, <\Sigma, \Phi>) = \Pi_{INSFH}(\Phi, \Sigma)$$

$$Symbols(\Pi^{ASL}_{INSFH}, <\Sigma, \Phi>) = \Sigma$$

$$\Gamma env(\Pi^{ASL}_{INSFH}, <\Sigma, \Phi>) = \Phi$$

$$Rules(\Pi^{ASL}_{INSFH}, SP_1 +_\Sigma SP_2) = inl''(Rules(\Pi^{ASL}_{INSFH}, SP_1)) \cup inr''(Rules(\Pi^{ASL}_{INSFH}, SP_2))$$

$$Symbols(\Pi^{ASL}_{INSFH}, SP_1 +_\Sigma SP_2) =$$

$$inl''(Symbols(\Pi^{ASL}_{INSFH}, SP_1)) \cup inr''(Symbols(\Pi^{ASL}_{INSFH}, SP_2))$$

$$\Gamma env(\Pi^{ASL}_{INSFH}, SP_1 +_\Sigma SP_2) =$$

$$inl''(\Gamma env(\Pi^{ASL}_{INSFH}, SP_1)) \cup inr''(\Gamma env(\Pi^{ASL}_{INSFH}, SP_2))$$

$$Rules(\Pi^{ASL}_{INSFH}, SP|_\Sigma) = Rules(\Pi^{ASL}_{INSFH}, SP)$$

$$Symbols(\Pi^{ASL}_{INSFH}, SP|_\Sigma) = Symbols(\Pi^{ASL}_{INSFH}, SP)$$

$$\Gamma env(\Pi^{ASL}_{INSFH}, SP|_\Sigma) = \Gamma env(\Pi^{ASL}_{INSFH}, SP)$$

$$Rules(\Pi^{ASL}_{INSFH}, rename \quad SP \quad \textbf{by} \quad \sigma) = \sigma''(Rules(\Pi^{ASL}_{INSFH}, SP))$$

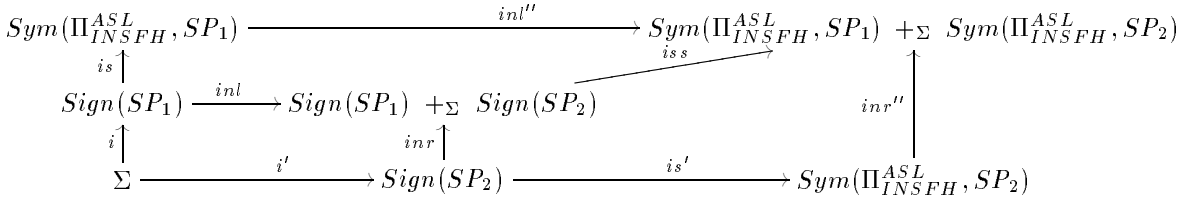
$$Symbols(\Pi^{ASL}_{INSFH}, rename \quad SP \quad \textbf{by} \quad \sigma) =$$

$$\sigma''(Symbols(\Pi^{ASL}_{INSFH}, SP))$$

$$\Gamma env(\Pi^{ASL}_{INSFH}, rename \quad SP \quad \textbf{by} \quad \sigma) =$$

$$\sigma''(\Gamma env(\Pi^{ASL}_{INSFH}, SP))$$

*where the overloaded symbols $inl''$ and $inr''$ are the pushout morphisms of the following diagram:*




*and $inl''$ and $inr''$ also denote the usual renaming functions between environments and proof systems which use the pushouts morphisms with the same name.*

*Since $Signature(SP_1) +_\Sigma Signature(SP_2)$ is a pushout $iss$ is the unique morphism with arity*

$$iss : Signature(SP_1) +_\Sigma Signature(SP_2) \hookrightarrow$$

$$Symbols(\Pi^{ASL}_{INSFH}, SP_1) +_\Sigma Symbols(\Pi^{ASL}_{INSFH}, SP_2)$$

*and the pushouts can be chosen in such a way that $iss$ is an inclusion.*

## 4.2 The proof system $\Pi^{ASL}_{FOLEQ}$ for the reachability and behavioural operators

In this subsection, we present the proof system $\Pi^{ASL}_{FOLEQ}$ for the reachability and behavioural operators. In this case, different proof rules and axioms are added to the proof system of the subspecification of each operator. First, we present some extensions of signatures which are needed to define some of the proof systems, then we present the proof rules with which the proof system of the reachability operator is extended and next we define the proof rules and axioms which are used to define the proof systems for the behavioural operators. Finally we present the proof system for first-order logic and the soundness and incompleteness results of $\Pi^{ASL}_{FOLEQ}$.

In the next definitions, we present different relational signatures which extend a given signature with the following symbols:

- Symbols to denote the observational equality for every sort of the signature and symbols to denote a definedness predicate also for every sort of the signature.
- The same extension as in the previous one plus symbols to denote a pseudoepimorphism between the original signature and a disjoint copy.
- The same extension as in the previous one plus symbols to denote contexts and context application.

**Definition 4.12** *The relational signature $\Sigma[\sim, D]$ is defined for any signature $\Sigma = (S, Op)$ and for any S-families of new symbols $\sim, D$ as:*

$$\Sigma[\sim, D] = \Sigma \cup \{\sim_s : s \times s | s \in S\} \cup \{D_s : s | s \in S\}$$

**Definition 4.13** *The relational signature* $\Sigma[\sim, D, \pi_{Copy}]$ *for any signature* $\Sigma = (S, Op)$*, for any bijective signature morphism* $Copy : \Sigma \to Copy(\Sigma)$ *such that* $\Sigma \cap Copy(\Sigma) = \emptyset$ *and for any* $S-$*family of new symbols* $\sim, D$ *and* $\pi$ *is defined as:*

$$\Sigma[\sim, D, \pi_{Copy}] = \Sigma[\sim, D] \cup Copy(\Sigma) \cup \{\pi_s : s \to Copy(s) | s \in S\}$$

**Remark:** *The relational signature* $Copy(\Sigma)[\sim, D, \pi_{Copy^{-1}}]$ *stands for the following signature:*

$$Copy(\Sigma)[\sim, D, \pi_{Copy^{-1}}] = Copy(\Sigma)[\sim, D] \cup \Sigma \cup \{\pi_s : Copy(s) \to s | s \in S\}$$

**Definition 4.14** *The relational signature* $\Sigma[\sim, D, \pi_{Copy}, z, c, c\_appl]$ *for any signature* $\Sigma = (S, Op)$*, for any bijective signature morphism* $Copy : \Sigma \to Copy(\Sigma)$ *such that* $\Sigma \cap Copy(\Sigma) = \emptyset$*, for any* $S-$*family of new symbols* $\sim$*,* $\pi$ *and* $z$ *and for any new symbols* $c$*,* $c\_appl$ *is defined as follows:*

$$\begin{aligned}
&Sorts(\Sigma[\sim, D, \pi_{Copy}, z, c, c_appl]) \;=\\
&\quad Sorts(\Sigma)[\sim, D, \pi_{Copy}] \;\cup\; \{c[r \to s] \mid r \in S, s \in S\}\\
&Ops(\Sigma[\sim, D, \pi_{Copy}, z, c, c_appl]) \;=\; Ops(\Sigma[\sim, D, \pi_{Copy}]) \;\cup\\
&\quad \{c[z_s] : c[s \to s] \mid s \in S\} \;\cup\\
&\quad \{c[r, f] : c[r \to s_1] \times \ldots \times c[r \to s_n] \to c[r \to s] \mid\\
&\qquad r \in S, f : s_1 \times \ldots \times s_n \to s \in Op\} \;\cup\\
&\quad \{c_appl[r, s] : c[r \to s] \times r \to s \mid r \in S, s \in S\}\\
&Prs(\Sigma[\sim, D, \pi_{Copy}, z, c, c_appl]) \;=\; Pr(\Sigma[\sim, D])
\end{aligned}$$

In the following, some proof rules to prove sentences of the form $\forall x : S.D_s(x) \supset \phi$ are presented:

**Definition 4.15** *The set of rules* $D\_sr[\Sigma[\sim, D], In]$ *is defined as follows:*

$$D\_sr[\Sigma[\sim, D], In] \;=\; \{D\_sr[\Sigma[\sim, D], In, s] \mid s \in In\}$$

*and the rule* $D\_sr[\Sigma[\sim, D], In, s]$ *is defined for every sort* $s \in S - In$ *as follows:*

$$\frac{\begin{array}{l}\{\Gamma \Rightarrow_X \forall x_1 : s_1. \ldots . \forall x_n : s_n.\\ \qquad \bigwedge_{s_i = s} \phi\,\{x_i \,/\, x\} \;\supset\; \phi\,\{f(x_1, \ldots, x_n) \,/\, x\} \mid\\ \qquad\qquad f : s_1 \times \ldots \times s_n \to s \in Ops(\Sigma)\}\end{array}}{\Gamma \Rightarrow_X \forall x : s.D_s(x) \supset \phi}$$

Next, proof rules to define the proof system associated to the reachability operator are presented:

**Definition 4.16** *The set of rules* $Reach\_sr[\Sigma, (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R})]$ *is defined as follows:*

$$Reach\_sr[\Sigma, (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R})] \;=\; \{Reach\_sr[\Sigma, (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R}), s] \mid s \in \mathcal{S}_\mathcal{R}\}$$

*and the rule* $Reach\_sr[\Sigma, (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R}), s]$ *is defined for every sort* $s \in \mathcal{S}_\mathcal{R}$ *as follows:*

$$\frac{\begin{array}{l}\{\Gamma \Rightarrow_X \forall x_1 : s_1. \ldots . \forall x_n : s_n. \\ \qquad \bigwedge_{s_i = s} \phi\,\{x_i \,/\, x\} \;\supset\; \phi\,\{f(x_1, \ldots, x_n) \,/\, x\} \;\mid \\ \qquad\qquad\qquad\qquad f : s_1 \times \ldots \times s_n \to s \in \mathcal{F}_\mathcal{R}\}\end{array}}{\Gamma \Rightarrow_X \forall x : s.\phi}$$

In the following, proof rules to derive proofs about observational equality are presented. The proof rules appear in the proof system of the behavioural operators and there are two different kind of proof rules:

- proof rules which define the observational equality.
- proof rules to perform context induction to reason about formulas of the form $\forall cxt : c[r \to s].\phi$

**Definition 4.17** *The set of rules* $Indist\_rel\_sr[\Sigma[\sim]]$ *is defined as follows:*

- *For each sort* $s \in S - Obs$*, the following rule:*

$$\frac{\begin{array}{l}\Gamma \Rightarrow_X \bigwedge_{obs \in Obs} \forall cxt : c[r \to obs]. \\ \qquad c\_appl[s, obs](cxt, t) \;=\; c\_appl[s, obs](cxt, r)\end{array}}{\Gamma \Rightarrow_X t \sim_s r}\; D_s(t) \& D_s(r)$$

- *The following context induction rule for each* $r \in S$ *and* $s \in S - In$ *such that* $r \neq s$*:*

$$\frac{\begin{array}{l}\{\Gamma \Rightarrow_X \forall cxt_1 : c[r \to s_1]. \ldots . \forall cxt_n : c[r \to s_n]. \\ \qquad \bigwedge_{s_i = s} \phi\,\{cxt_i / cxt\} \;\supset \phi\,\{c[r, f](cxt_1, \ldots, cxt_n)/cxt\} \;\mid \\ \qquad\qquad\qquad\qquad f : s_1 \times \ldots \times s_n \to s \in Op\}\end{array}}{\Gamma \Rightarrow_X \forall cxt : c[r \to s].\phi}$$

- *The following context induction rule for each* $r \in S$ *and* $s \in S - In$ *such that* $r = s$*:*

$$\frac{\begin{array}{l}\{\Gamma \Rightarrow_X \forall cxt_1 : c[r \to s_1]. \ldots . \forall cxt_n : c[r \to s_n].\, \phi\,\{c[z_s]/cxt\} \wedge \\ \qquad \bigwedge_{s_i = s} \phi\,\{cxt_i / cxt\} \;\supset \phi\,\{c[r, f](cxt_1, \ldots, cxt_n)/cxt\} \;\mid \\ \qquad\qquad\qquad\qquad f : s_1 \times \ldots \times s_n \to s \in Op\}\end{array}}{\Gamma \Rightarrow_X \forall cxt : c[r \to s].\phi}$$

- *For each sort $s \in Obs$, the following rule:*

$$\frac{\Gamma \vdash_X t = r}{\Gamma \vdash_X t \sim_s r} \; D_s(t) \& D_s(r)$$

In the following, you will find the axiomatisation of the pseudoepimorphism used in the proof system for the behavioural operator. This axiomatisation is formulated over the signatur $\Sigma[\sim, D, \pi_{Copy}]$ and it contains the following axioms:

- axioms to determine that $\pi_{Copy}$ is an homomorphism.
- axioms to determine that the homomorphism $\pi_{Copy}$ is surjective.
- axioms to establish the compatibility between the observational equality for every sort of the signature and the set theoretical equality associated to the disjoint copy of the given sort.

**Definition 4.18** *The set of sentences $pEpi_{FOLEQ}[\Sigma[\sim, D, \pi_{Copy}]]$ is defined as:*

$$pEpi_{FOLEQ}[\Sigma[\sim, D, \pi_{Copy}], Obs, In] = \; Hom[\Sigma[\sim, D, \pi_{Copy}], Obs, In] \; \cup$$

$$Epihom[\Sigma[\sim, D, \pi_{Copy}], Obs, In] \; \cup \; \sim -comp[\Sigma[\sim, D, \pi_{Copy}], Obs, In]$$

*where*

$$Hom[\Sigma[\sim, \pi_{Copy}], Obs, In] = \bigcup_{f: s_1 \times \ldots \times s_n \to s \in Op} \{\forall t_1 : s_1 \ldots . \forall t_n : s_n .$$

$$\bigwedge_{i \in [1..n], s_i \in S - In} D_{s_i}(t_i) \; \Rightarrow$$

$$\pi_{Copy,s}(f(t_1, \ldots, t_n)) = Copy(f)(\pi_{Copy,s_1}(t_1), \ldots, \pi_{Copy,s_n}(t_n))\}$$

$$Epihom[\Sigma[\sim, \pi_{Copy}], Obs, In] =$$

$$\bigcup_{s \in S - In} \{\forall y : Copy(s) . \exists x : s . D_s[\Sigma, In](x) \; \wedge \; \pi_{Copy,s}(x) = y)\}$$

$$\bigcup_{s \in In} \{\forall y : Copy(s) . \exists x : s . \; \pi_{Copy,s}(x) = y)\}$$

$$\sim -comp[\Sigma[\sim, \pi_{Copy}], Obs, In] =$$

$$\bigcup_{s \in S - In} \{\forall x : s . \forall y : s . D_s[\Sigma, In](x) \wedge D_s[\Sigma, In](y) \Rightarrow$$

$$x \sim_s y \Leftrightarrow \pi_{Copy,s}(x) \; = \; \pi_{Copy,s}(y)\}$$

$$\bigcup_{s \in In} \{\forall x : s . \forall y : s . x \sim_s y \Leftrightarrow \pi_{Copy,s}(x) \; = \; \pi_{Copy,s}(y)\}$$

Finally, we present an axiomatisation of the function application between contexts and values.

**Definition 4.19** *The set of sentences* $Ax\_c\_appl[\Sigma[\sim, D, \pi_{Copy}, z, c, c\_appl]]$ *is defined as:*

$$Ax\_c\_appl[\Sigma[\sim, D, \pi_{Copy}, z, c, c\_appl]] \ =$$

$$\{Ax\_c\_appl[\Sigma[\sim, D, \pi_{Copy}, z, c, c\_appl], r, s] \mid r \in S, s \in S\}$$

*where*

$$Ax\_c\_appl[\Sigma[\sim, D, \pi_{Copy}, z, c, c\_appl], r, s] \ =$$

$$\begin{cases} \forall v : r. c\_appl[r,s](c[z_r], v) \ = \ v \wedge c\_appl[r,s](x,v) \ = \ x \wedge \\ \bigwedge_{f:s_1 \times \ldots \times s_n \to s \in Op} \forall cxt_1 : c[r \to s_1]. \ldots . \forall cxt_n : c[r \to s_n]. \forall v : r. \\ \quad c\_appl[r,s](c[r,f](cxt_1, \ldots, cxt_n), v) = \\ \quad\quad f(c\_appl[r,s_1](cxt_1, v), \ldots, c\_appl[r,s_n](cxt_n, v)) & , if\ r = s \\ \\ \forall v : r.\ c\_appl[r,s](x,v) \ = \ x \wedge \\ \bigwedge_{f:s_1 \times \ldots \times s_n \to s \in Op} \forall cxt_1 : c[r \to s_1]. \ldots . \forall cxt_n : c[r \to s_n]. \forall v : r. \\ \quad c\_appl[r,s](c[r,f](cxt_1, \ldots, cxt_n), v) = \\ \quad\quad f(c\_appl[r,s_1](cxt_1, v), \ldots, c\_appl[r,s_n](cxt_n, v)) & , if\ r \neq s \end{cases}$$

**Definition 4.20** *The proof system* $\Pi^{ASL}_{FOLEQ}$ *is inductively defined for the specific operators of the language as follows:*

$$Rules(\Pi^{ASL}_{FOLEQ}, \textbf{reach} \quad SP \quad \textbf{with} \quad (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R})) \ = \ Rules(\Pi^{ASL}_{FOLEQ}, SP) \cup$$

$$Reach\_sr[Signature(SP), (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R})]$$

$$Symbols(\Pi^{ASL}_{FOLEQ}, \textbf{reach} \quad SP \quad \textbf{with} \quad (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R})) \ =$$

$$Symbols(\Pi^{ASL}_{FOLEQ}, SP)$$

$$\Gamma env(\Pi^{ASL}_{FOLEQ}, \textbf{reach} \quad SP \quad \textbf{with} \quad (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R})) \ = \ \Gamma env(\Pi^{ASL}_{FOLEQ}, SP)$$

$$Rules(\Pi^{ASL}_{FOLEQ}, \textbf{behaviour}\quad SP\quad \textbf{wrt}\quad \approx_{Obs,In}) =$$

$$Copy''(Rules(\Pi^{ASL}_{FOLEQ}, SP)) \cup$$

$$Indist\_rel\_sr[\Sigma[\sim], Obs, In]\ \cup\ D\_sr[\Sigma[\sim, D], In]$$

$$Symbols(\Pi^{ASL}_{FOLEQ}, \textbf{behaviour}\quad SP\quad \textbf{wrt}\quad \approx_{Obs,In}) =$$

$$Copy''(Symbols(\Pi^{ASL}_{FOLEQ}, SP)) \cup \Sigma[\sim, \pi_{Copy}, z, c, c\_appl]$$

$$\Gamma env(\Pi^{ASL}_{FOLEQ}, \textbf{behaviour}\quad SP\quad \textbf{wrt}\quad \approx_{Obs,In}) =$$

$$\Gamma env(rename\quad SP\quad \textbf{by}\quad Copy) \cup$$

$$pEpi_{FOLEQ}[\Sigma[\sim, \pi_{Copy}, z, c, c\_appl]]\ \cup$$

$$Ax\_c\_appl[\Sigma[\sim, \pi_{Copy}, z, c, c\_appl]] >)$$

$$Rules(\Pi^{ASL}_{FOLEQ}, \textbf{abstract}\quad SP\quad \textbf{by}\quad \equiv_{Obs,In}) =$$

$$Rules(\Pi^{ASL}_{FOLEQ}, \textbf{behaviour}\quad SP/\approx_{Obs,In}\quad \textbf{wrt}\quad \approx)$$

$$Symbols(\Pi^{ASL}_{FOLEQ}, \textbf{abstract}\quad SP\quad \textbf{by}\quad \equiv_{Obs,In}) =$$

$$Symbols(\Pi^{ASL}_{FOLEQ}, \textbf{behaviour}\quad SP/\approx_{Obs,In}\quad \textbf{wrt}\quad \approx)$$

$$\Gamma env(\Pi^{ASL}_{FOLEQ}, \textbf{abstract}\quad SP\quad \textbf{by}\quad \equiv_{Obs,In}) =$$

$$\Gamma env(\Pi^{ASL}_{FOLEQ}, \textbf{behaviour}\quad SP/\approx_{Obs,In}\quad \textbf{wrt}\quad \approx)$$

$$Rules(\Pi^{ASL}_{FOLEQ}, SP/\approx_{Obs,In}) =$$

$$Rules(\Pi^{ASL}_{FOLEQ}, rename \quad SP \quad \textbf{by} \quad Copy) \cup$$

$$Indist\_rel\_sr[Copy(\Sigma)[\sim, \pi_{Copy-1}, copyz, copyc, copyc\_appl],$$

$$Copy(Obs), Copy(In)] \ \cup \ D\_sr[Copy(\Sigma)[\sim, D], Copy(In)]$$

$$Symbols(\Pi^{ASL}_{FOLEQ}, SP/\approx_{Obs,In}) =$$

$$Copy''(Symbols(SP, \Pi^{ASL}_{FOLEQ})) \cup$$

$$Copy(\Sigma)[\sim, \pi_{Copy-1}, copyz, copyc, copyc\_appl]$$

$$\Gamma env(\Pi^{ASL}_{FOLEQ}, SP/\approx_{Obs,In}) =$$

$$\Pi^{ASL}_{FOLEQ}(rename \quad SP \quad \textbf{by} \quad Copy) \cup$$

$$pEpi_{FOLEQ}[Copy(\Sigma)[\sim, \pi_{Copy-1}, copyz, copyc, copyc\_appl],$$

$$Copy(Obs), Copy(In)]] \ \cup$$

$$Ax\_c\_appl[Copy(\Sigma)[\sim, \pi_{Copy-1}, copyz, copyc, copyc\_appl],$$

$$Copy(Obs), Copy(In)]] >$$

*where $\Sigma$ is the signature of the argument specification $SP$ for every case and the overloaded function symbol $\sigma''$ is used here as the pushout of the following diagram:*

$$\begin{array}{ccc} Sym(\Pi^{ASL}_{FOLEQ}, SP) & \xrightarrow{\sigma''} & PO(i, \sigma'') \\ {\scriptstyle i}\uparrow & & \uparrow \\ \Sigma & \xrightarrow{\quad\quad\sigma\quad\quad} & \Sigma' \end{array}$$

*where $i : \Sigma \hookrightarrow Sym(\Pi^{ASL}_{FOLEQ}, SP)$ and it is also used as the renaming function of environments and of the proof system $\Pi^{ASL}_{FOLEQ}(SP)$ which use the pushout just described in the obvious and standard way, and the overloaded symbol $Copy''$ is the pushout morphism of the following diagram:*

$$\begin{array}{ccc} Sym(\Pi^{ASL}_{FOLEQ}, SP) & \xrightarrow{Copy''} & PO(i, Copy) \\ {\scriptstyle i}\uparrow & & \uparrow \\ \Sigma & \xrightarrow{\quad Copy \quad} & Copy(\Sigma) \end{array}$$

*where* $i : \Sigma \hookrightarrow Symbols(\Pi^{ASL}_{FOLEQ}, SP)$ *and*

$$Copy''\ (Symbols\ (\Pi^{ASL}_{FOLEQ}, SP)) \cap$$

$$\Sigma[\sim, \pi_{Copy}, z, c, c\_appl] = \ \ Copy(\Sigma)$$

*and also*

$$Copy''\ (Symbols(\Pi^{ASL}_{FOLEQ}, SP)) \ \ \cap\ Copy(\Sigma)$$

$$[\sim, \pi_{Copy-1}, copyz, copyc, copyc\_appl] \ = \ Copy(\Sigma)$$

*The symbol* $Copy''$ *is also used as the renaming functions of environmnents and of set of rules using the pushout morphism just described with the same name in the obvious and standard way.*

*Note that although the proof system for the behaviour and quotient operators is defined in terms of the proof system of a structured specification expression, it is not needed to apply to the symbols of these proof systems the pushouts to avoid name clashes in structured specifications since the conditions which satisfy the pushout* $Copy''$ *guarantee no name clashes.*

Now, we present a proof system for first-order logic.

**Definition 4.21** *The natural deduction system* $\Pi_{FOLEQ}(\Gamma)$ *is defined by the following set of rules for any* $\Gamma \in \mathcal{P}(|Sen_{AINS}(\Sigma)|)$*:*

$$\frac{}{\{\} \Rightarrow_X \mathbf{true}} \quad (T) \qquad \frac{}{\Gamma \Rightarrow_X \mathbf{false} \supset \phi} \quad (F)$$

$$\frac{\Gamma \Rightarrow_X \phi_1 \wedge \phi_2}{\Gamma \Rightarrow_X \phi_1} \quad (\wedge El) \qquad \frac{\Gamma \Rightarrow_X \phi_1 \wedge \phi_2}{\Gamma \Rightarrow_X \phi_2} \quad (\wedge Er)$$

$$\frac{\Gamma \Rightarrow_X \phi_1 \quad \Gamma \Rightarrow_X \phi_2}{\Gamma \Rightarrow_X \phi_1 \wedge \phi_2} \quad (\wedge I)$$

$$\frac{\Gamma \Rightarrow_X \phi_1}{\Gamma \Rightarrow_X \phi_1 \vee \phi_2} \quad (\vee Il) \qquad \frac{\Gamma \Rightarrow_X \phi_2}{\Gamma \Rightarrow_X \phi_1 \vee \phi_2} \quad (\vee Ir)$$

$$\frac{\Gamma \Rightarrow_X \phi_1 \vee \phi_2 \quad \Gamma \Rightarrow_X \phi_1 \supset \psi \quad \Gamma \Rightarrow_X \phi_2 \supset \psi}{\Gamma \Rightarrow_X \psi} \quad (\vee E)$$

$$\frac{\Gamma \cup \{\phi\} \Rightarrow_X \mathbf{false}}{\Gamma \Rightarrow_X \neg\phi} \quad (\neg I) \qquad \frac{\Gamma \Rightarrow_X \psi \quad \Gamma \Rightarrow_X \neg\psi}{\Gamma \Rightarrow_X \phi} \quad (\neg E)$$

$$\frac{\Gamma \cup \phi \Rightarrow_X \phi'}{\Gamma \Rightarrow_X \phi \supset \phi'} \quad (\supset\ \ i) \qquad \frac{\Gamma \Rightarrow_X \phi \supset \phi' \quad \Gamma \Rightarrow_X \phi}{\Gamma \Rightarrow_X \phi'} \quad (\supset\ \ e)$$

$$\frac{\Gamma \Rightarrow_X \phi[t/x]}{\Gamma \Rightarrow_X \exists x : s.\phi}\ t \in T_{\Sigma,s}(X) \qquad (\exists I)$$

$$\frac{\Gamma \Rightarrow_X \exists x : s.\phi \quad \Gamma \cup \{\phi\} \Rightarrow_{X \cup \{x:s\}} \psi}{\Gamma \Rightarrow_X \psi} \qquad (\exists E)$$

$$\frac{\Gamma \Rightarrow_{X \cup \{x:s\}} \phi}{\Gamma \Rightarrow_X \forall x : s.\phi} \qquad (\forall\ I)$$

$$\frac{\Gamma \Rightarrow_X \forall x : s.\phi}{\Gamma \Rightarrow_X \phi\{t/x\}}\ t \in T_{\Sigma,s}(X) \qquad (\forall\ E)$$

$$\frac{}{\Gamma \Rightarrow_X t = t} \quad (REFL) \qquad \frac{\Gamma \Rightarrow_X r = s \quad \Gamma \Rightarrow_X s = t}{\Gamma \Rightarrow_X r = t} \qquad (TRANS)$$

$$\frac{\Gamma \Rightarrow_X t = s}{\Gamma \Rightarrow_X s = t} \quad (SYM) \qquad \frac{\Gamma \Rightarrow_{X \cup \{x:s'\}} \phi[s/x] \quad \Gamma \Rightarrow_X t = s}{\Gamma \Rightarrow_{X \cup \{x:s'\}} \phi[t/x]} \qquad (SUBST)$$

And next, we present the soundness and incompleteness results of $\Pi^{ASL}_{FOLEQ}$.

**Theorem 4.1** *The consequence relation* $\vdash_{\Pi^{ASL}_{FOLEQ}}$ *is sound.*

**Proof:**

The proof is by induction on specification expressions. For the two common operators the proof is trivial since $\vdash_{\Pi_{FOLEQ}(\emptyset)}$ is sound. For the rest of the cases, we have to prove the following propositions:

- For all specification expressions $SP \in SPEX(ASL)$ such that $Signature(SP) = \Sigma = (S, Op)$, if $\vdash_{\Pi^{ASL}_{FOLEQ}(SP)}$ is sound then $\vdash_{\Pi^{ASL}_{FOLEQ}(SP|_\Sigma)}$ is sound which follows trivially.
- For all specification expressions $SP \in SPEX(ASL)$ such that $Signature(SP) = \Sigma = (S, Op)$, if $\vdash_{\Pi^{ASL}_{FOLEQ}(SP)}$ is sound then $\vdash_{\Pi^{ASL}_{FOLEQ}(rename \quad SP \quad \mathbf{by} \quad \sigma)}$ is sound where $\sigma : Signature(SP) \rightarrow \Sigma'$.
  Assume that $SP \in SPEX(ASL)$ and assume that $\vdash_{\Pi^{ASL}_{FOLEQ}(SP)}$ is sound. We have to prove that for any $A \in Models(rename \quad SP \quad \mathbf{by} \quad \sigma)$ there exists a $Symbols(\Pi^{ASL}_{FOLEQ}(rename \quad SP \quad \mathbf{by} \quad \sigma))$-algebra (which we will refer as $A'$) such that:
  - $A'|_{\Sigma'} = A$.
  - $\forall \psi \in \Gamma env(\Pi^{ASL}_{AINS}, rename \quad SP \quad \mathbf{by} \quad \sigma).A' \models \psi$.
  - $A'$ satisfies $\Pi^{ASL}_{AINS}(rename \quad SP \quad \mathbf{by} \quad \sigma)$.

Since $A|_\sigma \in Models(SP)$, by soundness of $\vdash_{\Pi^{ASL}_{FOLEQ}(SP)}$, we know that there exists a $Symbols(\Pi^{ASL}_{FOLEQ}(SP)$-algebra which we will refer as $A''$ such that:

- $A''|_\Sigma = A|_\sigma$.
- $\forall \psi \in \Gamma env(\Pi^{ASL}_{AINS}, SP).A' \models \psi$.
- $A''$ satisfies $\Pi^{ASL}_{AINS}(SP)$.

It is straightforward to prove that the $Symbols(\Pi^{ASL}_{FOLEQ}$ $(rename \quad SP \quad \textbf{by} \quad \sigma))$ -algebra $A' = A''|_{(i;\sigma'')-1}$ where $\sigma''$ is the pushout morphism of $i : \Sigma \hookrightarrow Symbols(\Pi^{ASL}_{AINS}(SP))$ and $\sigma : \Sigma \to \Sigma'$ of the definition the proof system for this operator, satisfies the soundness conditions of the satisfaction condition of $AINS$ and because $A''$ satisfies the soundness conditions for the proof system $\Pi^{ASL}_{AINS}(SP)$.

- For all specification expressions $SP \in SPEX(ASL)$ such that $Signature(SP) = \Sigma = (S, Op)$, if $\vdash_{\Pi^{ASL}_{FOLEQ}(SP)}$ is sound then $\vdash_{\Pi^{ASL}_{FOLEQ}(\textbf{reach} \quad SP \quad \textbf{with} \quad (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R}))}$ is sound.
  Assume that $SP \in SPEX(ASL)$ and assume that $\vdash_{\Pi^{ASL}_{FOLEQ}(SP)}$ is sound.
  To prove that $\vdash_{\Pi^{ASL}_{FOLEQ}(\textbf{reach} \quad SP \quad \textbf{with} \quad (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R}))}$ is sound, we have to define for any $A \in Models(\textbf{reach} \quad SP \quad \textbf{with} \quad (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R}))$,
  a $Symbols(\Pi^{ASL}_{FOLEQ}(\textbf{reach} \quad SP \quad \textbf{with} \quad (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R})))$-algebra $B$ which satisfies the soundness conditions.

  Assume that $A \in Models(\textbf{reach} \quad SP \quad \textbf{with} \quad (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R}))$. By the semantics of the reachability operator, we know also that $A \in Models(SP)$.

  Since $\Pi^{ASL}_{FOLEQ}(SP)$ is sound, we know that there exists a $Symbols(\Pi^{ASL}_{FOLEQ}(SP))$-algebra $A'$ such that $A'$ satisfies the soundness conditions and therefore $A'|_{Signature(SP)} = A$.

  Assume that $B$ is a $Symbols(\Pi^{ASL}_{FOLEQ}(SP))$-algebra which satisfies the soundness conditions and $B'|_{Signature(SP)} = A$. To prove that $B$ satisfies the soundness conditions we have to show that:

  - $B|_{Signature(SP)} =$ A which obviously holds
  - $\forall \psi \in \Gamma env(\Pi^{ASL}_{AINS}, \textbf{reach} \quad SP \quad \textbf{with} \quad (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R})).B \models \psi$ which holds since $\Gamma env(\Pi^{ASL}_{AINS}, \textbf{reach} \quad SP \quad \textbf{with} \quad (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R})) = \Gamma env(\Pi^{ASL}_{AINS}, SP)$.
  - B satisfies $\Pi^{ASL}_{FOLEQ}(\textbf{reach} \quad SP \quad \textbf{with} \quad (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R}))$ because B satisfies
    $\Pi^{ASL}_{FOLEQ}(SP)$ and for each sort $s \in \mathcal{S}_\mathcal{R}$ B satisfies $Reach\_sr[\Sigma, (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R}), s]$ and it follows trivially by term induction on the carrier set $B_s$ since $B|_{Signature(SP)} = A$ satisfies the reachability constraint.

- For all specification expressions $SP \in SPEX(ASL)$ such that $Signature(SP) = \Sigma = (S, Op)$, if $\vdash_{\Pi^{ASL}_{FOLEQ}(SP)}$ is sound then $\vdash_{\Pi^{ASL}_{FOLEQ}(\textbf{behaviour} \quad SP \quad \textbf{wrt} \quad \approx_{Obs,In})}$ is sound.
  Assume that $SP \in SPEX(ASL)$ and assume that $\vdash_{\Pi^{ASL}_{FOLEQ}(SP)}$ is sound.
  To prove that $\vdash_{\Pi^{ASL}_{FOLEQ}(\textbf{behaviour} \quad SP \quad \textbf{wrt} \quad \approx_{Obs,In})}$ is sound, we have to define for any $A \in Models(\textbf{behaviour} \quad SP \quad \textbf{wrt} \quad \approx_{Obs,In})$, a $Symbols(\Pi^{ASL}_{FOLEQ}(\textbf{behaviour} \quad SP \quad \textbf{wrt} \quad \approx_{Obs,In}))$-algebra $B$ such that:

  - $B|_{Signature(SP)} = $ A
  - $\forall \psi \in \Gamma env(\Pi^{ASL}_{AINS}, SP).B \models \psi$.
  - B satisfies $\Pi^{ASL}_{FOLEQ}(\textbf{behaviour} \quad SP \quad \textbf{wrt} \quad \approx_{Obs,In})$

  Since $\Pi^{ASL}_{FOLEQ}(SP)$ is sound, there exists a $Symbols(\Pi^{ASL}_{FOLEQ}(SP))$-algebra $A'$ such that $A'|_{Signature(SP)} = A/\approx^{In,Obs}_{A}$ and $A'$ satisfies $\Pi^{ASL}_{FOLEQ}(SP)$.
  The $Symbols(\Pi^{ASL}_{FOLEQ}(\textbf{behaviour} \quad SP \quad \textbf{wrt} \quad \approx_{Obs,In}))$-algebra $B$ is defined as follows:

  $B|_{Signature(SP)} = $ A ( and therefore it satisfies the first condition).

  $B|_{Copy''(Symbols(\Pi^{ASL}_{FOLEQ}(SP)))} = A'|_{Copy''-1}$

  $B|_{\pi_s} = \epsilon_{A,s}$for every sort $s \in S$.

  *where*

  $\epsilon_{A,s}(v) = [v]^{Obs,In}_{\approx_{A,s}}$ for all $v \in A_s$.

$B|_{\sim_s} = \approx_{A,s}^{In,Obs}$ for every sort $s \in S$.

$B|_{c[r,s]} = \{\lambda frv : A[In]_r.I_{\alpha \cup \{(z_r, frv)\}}(czr) \mid$

$\alpha \in (X_{In} \to A)$ , $czr \in T_{\Sigma,s}(X_{In} \cup \{z_r\})\}$for every sort $s, r \in S$

and for a $S$-sorted infinite enumerable set of variables such that

for every sort $s \in S$ $X_s \cap z_s = \emptyset$.

$c[z_s]_B = z_s$ for every sort $s \in S$

$c[r,f]_B(cx_1, \ldots, cx_n) = f(cx_1, \ldots, cx_n)$ for every sort $r \in S$,

$f : s_1 \times \ldots \times s_n \to s \in Op$,

$cx_1 \in T_{\Sigma,s_1}(X_{In} \cup \{z_r\})$, ... , $cx_n \in T_{\Sigma,s_n}(X_{In} \cup \{z_r\})$

$c\_appl[r,s]_B(\lambda frv : A[In]_r.I_{\alpha \cup \{(z_r, frv)\}}(czr), rv) =$

$I_{\alpha \cup \{(z_r, rv)\}}(czr)$ for all $\alpha : X_{In} \to A[In]$,

for all $czr \in T_{\Sigma,s}(X_{In} \cup \{z_r\})$ and for all $rv \in A[In]_s$

To show that

$$\forall \psi \in \Gamma env(\Pi_{AINS}^{ASL}, \textbf{behaviour} \quad SP \quad \textbf{wrt} \quad \approx_{Obs,In}).B \models \psi$$

we have to prove the following propositions:

- $\forall \psi \in \Gamma env(\Pi_{AINS}^{ASL}, rename \quad SP \quad \textbf{by} \quad Copy).B \models \psi$ which follows by the induction hypotheses and the proof of soundness for the rename operator.
- $B \models pEpi_{FOLEQ}[\Sigma[\sim, \pi_{Copy}, z, c, c\_appl]] \wedge$

  $B \models Ax\_c\_appl[\Sigma[\sim, \pi_{Copy}, z, c, c\_appl]]$

  $B \models pEpi_{FOLEQ}[\Sigma[\sim, \pi_{Copy}, z, c, c\_appl]]$ because for every sort $s$ $B_{\pi_s}$ can be seen as an epimorphism between $A$ and $A/\approx_{A,s}^{Obs,In}$ and

therefore $B$ satisfies $Hom[\Sigma[\pi_{Copy}],\ Epihom[\Sigma[\pi_{Copy}]]$ and also $\sim -comp[\Sigma[\sim, \pi_{Copy}]]$. The proof of $B \models Ax\_c\_appl[\Sigma[\sim, \pi_{Copy}, z, c, c\_appl]]$ is straightforward.

And to show that $B$ satisfies $\Pi^{ASL}_{FOLEQ}(\textbf{behaviour}\quad SP\quad \textbf{wrt}\quad \approx_{Obs,In})$ we have to show the following propositions:

- $B|_{Copy''(Symbols(\Pi^{ASL}_{FOLEQ}(SP)))}$ satisfies $\Pi^{ASL}_{FOLEQ}(rename\quad SP\quad \textbf{by}\quad Copy)$.
  Since $A \in Models(\textbf{behaviour}\quad SP\quad \textbf{wrt}\quad \approx_{Obs,In})$, we have that $A/\approx^{Obs,In}_{A} \in Models\ (SP)$ and by induction hypotheses we have that $A'$ satisfies $\Pi^{ASL}_{FOLEQ}(SP)$ where $A'|_{Signature(SP)} = A/\approx^{Obs,In}_{A}$.
  Therefore,
  $B|_{Copy''(Symbols(\Pi^{ASL}_{FOLEQ}(SP)))}$ satisfies $\Pi^{ASL}_{FOLEQ}\ (rename\quad SP\quad \textbf{by}\quad Copy)$.
- $B$ satisfies $Indist\_rel\_sr[\Sigma[\sim, \pi_{Copy}, z, c, c\_appl], Obs, In]$.
  $B$ satisfies

$$\frac{\Gamma \Rightarrow_X \bigwedge_{obs \in Obs} \forall cxt : c[r \to obs].\ c\_appl[r, obs](cxt, t) \ =\ c\_appl[r, obs](cxt, r)}{\Gamma \Rightarrow_X t \sim_s r}\ D_s\left[\Sigma, In\right](t) \& D_s\left[\Sigma, In\right](r)$$

  for each sort $s \in S - Obs$ and

$$\frac{\Gamma \vdash_X t = r}{\Gamma \vdash_X t \sim_s r}\ D_s\left[\Sigma, In\right](t) \& D_s\left[\Sigma, In\right](r)$$

  for each sort $s \in S$ because $B|_{\sim} = \approx^{In,Obs}_{A,s}$.
  $B$ satisfies

$$\frac{\begin{array}{l}\{\Gamma \Rightarrow_X \forall cxt_1 : c[r \to s_1]. \ldots . \forall cxt_n : c[r \to s_n].\ \bigwedge_{s_i = s} \phi\ \{cxt_i/cxt\}\ \supset \phi\ \{c[r, f](cxt_1, \ldots, cxt_n)/cxt\}\ \mid \\ \qquad\qquad f : s_1\ \times\ \ldots\ \times\ s_n\ \to\ s \in Op\}\end{array}}{\Gamma \Rightarrow_X \forall cxt : c[r \to s].\phi}$$

  by induction on the context $cxt : c[r \to s]$

- For all specification expressions $SP \in SPEX(ASL)$ such that $Signature(SP) = \Sigma = (S, Op)$, if $\vdash_{\Pi^{ASL}_{FOLEQ}(SP)}$ is sound then $\vdash_{\Pi^{ASL}_{FOLEQ}(SP/\approx_{Obs,In})}$ is sound.
  Assume that $SP \in SPEX(ASL)$ and assume that $\vdash_{\Pi^{ASL}_{FOLEQ}(SP)}$ is sound.
  To prove that $\vdash_{\Pi^{ASL}_{FOLEQ}(SP/\approx_{Obs,In})}$ is sound, we have to define for any $A \in Models(SP/\ \approx_{Obs,In})$, a $Symbols(\Pi^{ASL}_{FOLEQ}(SP/\ \approx_{Obs,In}))$-algebra $B$ such that:
  - $B|_{Signature(SP)} = \mathrm{A}$
  - $\forall \psi \in \Gamma env(\Pi^{ASL}_{AINS}, SP/\ \approx_{Obs,In}).B \models \psi$.

– B satisfies $\Pi^{ASL}_{FOLEQ}(SP/\approx_{Obs,In})$

By the definition of $Models(SP/\approx_{Obs,In})$, we know that there exists a $Signature(SP)$-algebra $A'$ such that $A'/\approx^{In,Obs}_{A'} \cong A$ and $A' \in Models(SP)$. In the following, we will denote by $h : A'/\approx^{Obs,In}_{A'} \to A$ the isomorphism between $A'/\approx^{Obs,In}_{A'}$ and $A$. By induction hypotheses, we know that there exists a $Symbols(\Pi^{ASL}_{FOLEQ}(SP))$-algebra $A''$ such that $A''|_{Signature(SP)} = A'$ and $A''$ satisfies $\Pi^{ASL}_{FOLEQ}(SP)$. The $Symbols(\Pi^{ASL}_{FOLEQ}(SP/\approx_{Obs,In}))$-algebra $B$ is defined as follows:

$B|_\Sigma = A$(and therefore it satisfies the first condition).

$B|_{Copy''(Symbols(\Pi^{ASL}_{FOLEQ}(SP)))} = A''|_{Copy''-1}$.

$B_{\pi_{Copy(s)}} = \epsilon_{A'|_{Copy-1},Copy(s)}$for every sort $s \in S$.

*where*

$\epsilon_{A,Copy(s)}(v) = h([v]_{\approx^{Copy(Obs),Copy(In)}_{A,Copy(s)}})$for any $A$ in $Alg(Copy(\Sigma))$,

for any $s \in S$ and for any $v \in A_{Copy(s)}$

$B_{\sim_{Copy(s)}} = \approx^{Copy(In),Copy(Obs)}_{A'|_{Copy-1},Copy(s)}$

$B|_{c[Copy(r),Copy(s)]} =$

$\{\lambda frv : A'|_{Copy-1}[Copy(In)]_{Copy(r)}.I_{\alpha\cup\{(z_{Copy(r)},frv)\}}(czr) \mid$

$\alpha \in (X_{Copy(In)} \to A')\ ,\ czr \in T_{Copy(\Sigma),Copy(s)}(X_{Copy(In)} \cup \{z_{Copy(r)}\})\}$

for every sort $s, r \in S$ and for a $S$-sorted infinite enumerable set of variables such that for every sort $s \in S$ $X_s \cap z_s = \emptyset$.

$c[z_{Copy(s)}]_B = z_{Copy(s)}$for every sort $s \in S$

$c[Copy(r), Copy(f)]_B(cx_1, \ldots, cx_n) = Copy(f)(cx_1, \ldots, cx_n)$

for every sort $r \in S$, $f : s_1 \times \ldots \times s_n \to s \in Op$ ,

$cx_1 \in T_{Copy(\Sigma),Copy(s_1)}(X_{Copy(In)} \cup \{z_{Copy(r)}\}), \ldots ,$

$cx_n \in T_{Copy(\Sigma),Copy(s_n)}(X_{Copy(In)} \cup \{z_{Copy(r)}\})$

$c\_appl[Copy(r), Copy(s)]_B$

$(\lambda frv : A'[Copy(In)]_{Copy(r)}.I_{\alpha \cup \{(z_{Copy(r)}, frv)\}}(czr), rv) \;\; =$

$I_{\alpha \cup \{(z_{Copy(r)}, rv)\}}(czr)$ for all $\alpha : X_{Copy(In)} \to A[Copy(In)]$,

for all $czr \in T_{Copy(\Sigma),Copy(s)}(X_{Copy(In)} \cup \{z_{Copy(r)}\})$

and for all $rv \in A[Copy(In)]_{Copy(s)}$ .

To show that

$$\forall \psi \in \Gamma env(\Pi^{ASL}_{AINS}, SP/\approx_{Obs,In}).B \models \psi$$

we have to show the following propositions:

- $\forall \psi \in \Gamma env(\Pi^{ASL}_{AINS}, rename \quad SP \quad \textbf{by} \quad Copy).B \models \psi$ which follows by the induction hypotheses and the proof of soundness for the rename operator.
- $B \models pEpi_{FOLEQ}[Copy(\Sigma)[\sim, \pi_{Copy-1}, z, c, c\_appl]] \wedge$

  $B \models Ax\_c\_appl[Copy(\Sigma)[\sim, \pi_{Copy-1}, z, c, c\_appl]]$

  $B \models pEpi_{FOLEQ}[Copy(\Sigma)[\sim, \pi_{Copy-1}, z, c, c\_appl]]$ because for every sort $s$ $B_{\pi_{Copy(s)}}$ can be seen as an epimorphism between $A'$ and $A$ and therefore $B$ satisfies $Hom[Copy(\Sigma)[\sim, \pi_{Copy}],\ Epihom[Copy(\Sigma)[\sim, \pi_{Copy}]]$ and also $\sim -comp[Copy(\Sigma)[\sim, \pi_{Copy}]]$. The proof of $B \models Ax\_c\_appl[\Sigma[\sim, \pi_{Copy}, z, c, c\_appl]]$ is straightforward.

And to show that $B$ satisfies $\Pi^{ASL}_{FOLEQ}(SP/\approx_{Obs,In})$ we have to show the following propositions:

- $B|_{Copy''(Symbols(\Pi^{ASL}_{FOLEQ}(SP)))}$ satisfies $\Pi^{ASL}_{FOLEQ}(rename \quad SP \quad \textbf{by} \quad Copy)$. Since $A' \in Models(SP)$, by induction hypotheses we have that $A''$ satisfies $\Pi^{ASL}_{FOLEQ}(SP)$ where $A''|_{Signature(SP)} = A'$. Therefore, $B|_{Copy''(Symbols(\Pi^{ASL}_{FOLEQ}(SP)))}$ satisfies $\Pi^{ASL}_{FOLEQ}(rename \quad SP \quad \textbf{by} \quad Copy)$.
- $B$ satisfies $Indist\_rel\_sr[Copy(\Sigma)[\sim, \pi_{Copy-1}, z, c, c\_appl], Copy(Obs), Copy(In)]$ in the same way as in the case of the behavioural operator.

- For all specification expressions $SP \in SPEX(ASL)$ such that $Signature(SP) = \Sigma = (S, Op)$, if $\vdash_{\Pi^{ASL}_{FOLEQ}(SP)}$ is sound then $\vdash_{\Pi^{ASL}_{FOLEQ}(\textbf{abstract} \quad SP \quad \textbf{by} \quad \equiv_{Obs,In})}$ is sound.
  It is trivial since by the two previous proofs we know that if $\vdash_{\Pi^{ASL}_{FOLEQ}(SP)}$ is sound then $\vdash_{\Pi^{ASL}_{FOLEQ}(\textbf{behaviour} \quad SP/\approx_{Obs,In} \quad \textbf{wrt} \quad \approx)}$ is sound.

**Theorem 4.2** *There is no sound and complete consequence relation of the form* $\vdash_{\Pi^{ASL}_{FOLEQ}}$ *for ASL.*

**Proof:**
See [6]. The basic idea of the proof is to define a specification $SP$ such that the set $\{\phi \mid SP \models \phi\}$ is not recursively enumerable since they work with consequence relations $\vdash$ which are recursively enumerable.They show that the set of sentences which satisfy the single sorted algebra $NAT$ with carrier set the natural numbers and with the usual operations $zero, succ, substract, add, multiply$ (where $substract_{NAT}(m, n) = 0$ if $n > m$) is not recursively enumerable. They use equational logic with data constraints to define an specification whose class of models is the class of algebras isomorphic to $NAT$. Since all the consequence relations of our proof systems are recursively enumerable because we work with natural deduction systems with a finite set of finitary rules, the proof is valid for our framework if we define the specification in ASL. We will refer to this specification as $NSP$ and it is defined as follows:

$$NSP = reach\ Nat\ with\ (nat, (zero : nat, succ : nat \to nat))$$

where

$$Nat = < (nat, zero : nat, succ : nat \to nat, add : nat \times nat \to nat,$$

$$substract : nat \times nat \to nat, multiply : nat \times nat \to nat),$$

$$(\forall m, n : nat.add(n, zero) = n \wedge add(n, succ(m)) = succ(add(n, m)) \wedge$$

$$(\forall m, n : nat.substract(zero, n) = zero \wedge$$

$$substract(succ(n), succ(m)) = substract(n, m)) \wedge$$

$$(\forall m, n : nat.multiply(n, zero) = zero \wedge$$

$$multiply(n, succ(m)) = add(n, multiply(n, m))$$

$$(\forall m, n : nat.\neg(n = n + succ(m))) >$$

It is trivial to show that

$$Models(NSP) = \{A \mid A \cong NAT\}$$

since one can prove that there exists an isomorphism between any model of $NSP$ and the initial model of $NSP$ which is isomorphic to $NAT$.

## 4.3 The proof system $\Pi^{ASL}_{HOL}$ for the reachability and behavioural operators

In this subsection, we present the proof system $\Pi^{ASL}_{HOL}$ for the reachability and behavioural operators. Before presenting the proof system, we present some extension of signatures which are needed to define the proof systems, next we present some axioms to define the proof system for the reachability operator and after that some axioms to define the proof systems for the behavioural operators. Finally, we present a proof system for higher-order logic and a proof of soundness and incompleteness of $\Pi^{ASL}_{HOL}$

In the next definitions, we present different relational signatures which extend a given signature with the following symbols:

- Symbols to denote the observational equality for every sort of the signature.
- The same extension as in the previous one plus symbols to denote a pseudoepimorphism between the original signature and a disjoint copy.

**Definition 4.22** *The relational signature* $\Sigma[\sim]$ *is defined for any signature* $\Sigma = (S, Op)$ *and for any S-family of new symbols* $\sim$ *as:*

$$\Sigma[\sim] = \Sigma \cup \{\sim_s : s \times s | s \in S\}$$

**Definition 4.23** *The relational signature* $\Sigma[\sim, \pi_{Copy}]$ *for any signature* $\Sigma = (S, Op)$*, for any bijective signature morphism* $Copy : \Sigma \to Copy(\Sigma)$ *such that* $\Sigma \cap Copy(\Sigma) = \emptyset$ *and for any* $S-$*family of new symbols* $\sim$ *and* $\pi$ *is defined as:*

$$\Sigma[\sim, \pi_{Copy}] = \Sigma[\sim] \cup Copy(\Sigma) \cup \{\pi_s : s \to Copy(s) | s \in S\}$$

**Remark:** *The relational signature* $Copy(\Sigma)[\sim, \pi_{Copy^{-1}}]$ *stands for the following signature:*

$$Copy(\Sigma)[\sim, \pi_{Copy^{-1}}] = Copy(\Sigma)[\sim] \cup \Sigma \cup \{\pi_s : Copy(s) \to s | s \in S\}$$

Here we present some axioms used to define the proof system for the reachability operator:

**Definition 4.24** *The set of sentences* $Reach\_ax[\Sigma, (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R})]$ *for any reachability constraint* $(\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R})$ *of* $\Sigma$ *is defined as follows:*

$$Reach\_ax[\Sigma, (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R})] = \{\forall P : [s].(\bigwedge_{f:s_1 \times \ldots \times s_n \to s \in \mathcal{F}_\mathcal{R}} (\forall x_1 : s_1 \ldots . \forall x_n : s_n.$$

$$(\bigwedge_{i \in [1..n], s_i = s} P(x_i) \supset P(f(x_1, \ldots, x_n))) \supset \forall x : s.P(x) \mid s \in \mathcal{F}_\mathcal{R} \}$$

In the following, there are some definitions which axiomatise the observational equality in higher-order logic. This axiomatisation contains the following axioms:

- Axioms to determine which a given value is reachable.
- Axioms to determine which a given relation coincide with the set-theoretical equality.
- Axioms to determine that a family of relations is a congruence.

As we mentioned in the introduction, the axiomatisation of the observational equality is like [5]. Basically, this axiomatisation is correct because the observational equality is the greates congruence which coincide with the set-theoretical equality for observable sorts.

**Definition 4.25** *The $Sorts(\Sigma)$-family of sentences $D[\Sigma, In]$ for any $In \subseteq Sorts(\Sigma)$ is defined as follows:*

$$D[\Sigma, In] = \{D_s[\Sigma, In] \mid s \in Sorts(\Sigma)\}$$

*where*

$$D_s[\Sigma, In] = \lambda x : s.(\forall P : [s].(\bigwedge_{f:s_1 \times \ldots \times s_n \to s \in Ops(\Sigma)} (\forall x_1 : s_1 \ldots . \forall x_n : s_n.$$

$$(\bigwedge_{i \in [1..n], s_i = s} P(x_i) \supset P(f(x_1, \ldots, x_n))))) \supset P\ x) \qquad , if\ s \in S - In$$

$$D_s[\Sigma, In] = \lambda x : s.\mathbf{true} \qquad , if\ s \in In$$

**Definition 4.26** *The set of sentences $Indist\_rel[(S, Op)[\sim], Obs, In]$ is defined as follows:*

$$Indist\_rel[(S, Op)[\sim], Obs, In] = \{\ Indist\_rel[(S, Op)[\sim], Obs, In, s]\ \mid\ s \in S\}$$

*where*

$$Indist\_rel[(S,Op)[\sim],Obs,In,s] = \lambda x : s.\lambda y : s.\exists_{s' \in S} R_{s'} : [s',s'].$$

$$R_s(x,y) \ \wedge OBSEQ[(S,Op),R,Obs,In] \ \wedge$$

$$CONG[(S,Op),R,Obs,In] \ \wedge \ D_s[\Sigma,In](x) \ \wedge \ D_s[\Sigma,In](y)$$

$$OBSEQ[(S,Op),R,Obs,In] = \bigwedge_{obs \in Obs} \forall v : obs.\forall w : obs.$$

$$D_{obs}[\Sigma,In](v) \ \wedge D_{obs}[\Sigma,In](w) \ \Rightarrow (R_{obs}(v,w) \Leftrightarrow v = w)$$

$$CONG[(S,Op),R,Obs,In] =$$

$$\bigwedge_{f:s_1 \times \ldots \times s_n \to s \in Op} \forall x_1 : s_1.\forall y_1 : s_1. \ \ldots \ .\forall x_n : s_n.\forall y_n : s_n.$$

$$(R_{s1}(x_1,y_1) \ \wedge \ \ldots \ \wedge \ R_{sn}(x_n,y_n) \ \Rightarrow$$

$$R_s(f(x_1,\ldots,x_n),f(y_1,\ldots,y_n)))$$

Here, we present an axiomatisation of a pseudoepimorphism over the signature $\Sigma[\sim,\pi_{Copy}]$ in higher-order logic. This axiomatisation is equivalent to the one presented in first-order logic.

**Definition 4.27** *The set of sentences* $pEpi_{HOL}[\Sigma[\sim,\pi_{Copy}],Obs,In]$ *is defined as:*

$$pEpi_{HOL}[\Sigma[\sim,\pi_{Copy}],Obs,In] \ = \ Hom[\Sigma[\sim,\pi_{Copy}],Obs,In] \cup$$

$$Epihom[\Sigma[\sim,\pi_{Copy}],Obs,In] \ \cup \ \sim -comp[\Sigma[\sim,\pi_{Copy}],Obs,In]$$

*where*

$$Hom[\Sigma[\sim,\pi_{Copy}],Obs,In] = \bigcup_{f:s_1 \times \ldots \times s_n \to s \in Op} \{\forall t_1 : s_1. \ldots .\forall t_n : s_n.$$

$$\bigwedge_{i \in [1..n]} D_{s_i}[\Sigma,In](t_i) \ \Rightarrow$$

$$\pi_{Copy,s}(f(t_1,\ldots,t_n)) = Copy(f)(\pi_{Copy,s_1}(t_1),\ldots,\pi_{Copy,s_n}(t_n))\}$$

$$Epihom[\Sigma[\sim, \pi_{Copy}], Obs, In] =$$

$$\bigcup_{s \in S} \{\forall y : Copy(s).\exists x : s.D_s[\Sigma, In](x) \ \wedge \ \pi_{Copy,s}(x) = y)\}$$

$$\sim -comp[\Sigma[\sim, \pi_{Copy}], Obs, In] =$$

$$\bigcup_{s \in S} \{\forall x : s.\forall y : s.D_s[\Sigma, In](x) \wedge D_s[\Sigma, In](y) \Rightarrow$$

$$x \sim_s y \Leftrightarrow \pi_{Copy,s}(x) \ = \ \pi_{Copy,s}(y)\}$$

In the following definitions, we define specification expressions using an extended version of $ASL$ with relational signatures and we prove some equivalences between the semantics of some of these specification expressions and the semantics of the behavioural operator. The semantics of $ASL$ with relational signatures is extended in the obvious way basically just replacing signatures by relational signatures. The specification expressions defined below are used to establish an equivalence between the semantics of behavioural operators and structured specifications which are the basis to define the proof system $\Pi^{ASL}_{HOL}$.

**Definition 4.28** *The specification expression $BSP[\Sigma[\sim, \pi_{Copy}], Obs, In]$ for any signature $\Sigma = (S, Op)$, for any bijective signature morphism $Copy : \Sigma \rightarrow Copy(\Sigma)$ such that $\Sigma \cap Copy(\Sigma) = \emptyset$ and for any $S-$family of new symbols $\sim$ and $\pi$ is defined as:*

$$BSP[\Sigma[\sim, \pi_{Copy}], Obs, In] =< \Sigma[\sim, \pi_{Copy}],$$

$$\{\forall x : s.\forall y : s.x \sim_s y \Leftrightarrow Indist\_rel[\Sigma[\sim], Obs, In, s](x, y) \ | \ s \in S\} \ \cup$$

$$pEpi[\Sigma[\sim, \pi_{Copy}], Obs, In] >$$

**Definition 4.29** *For any specification expression $SP$ with signature $\Sigma = (S, Op)$, for any signature morphism $Copy : \Sigma \rightarrow Copy(\Sigma)$ and $Copy' : \Sigma \rightarrow Copy'(\Sigma)$ such that $\Sigma \cap Copy(\Sigma) = \emptyset$ and $\Sigma \cap Copy'(\Sigma) = \emptyset$, for any $S-$family of new symbols $\sim$ and $\pi$, the specification expression $GBSP[SP, Copy, \Sigma[\sim, \pi_{Copy'}]]$ is defined as:*

$$GBSP[SP, Copy, \Sigma[\sim, \pi_{Copy'}], Obs, In] =$$

$$rename \quad SP \quad \textbf{by} \quad Copy \ + \ BSP[\Sigma[\sim, \pi_{Copy'}], Obs, In]$$

**Lemma 4.1** *Let $R_A^{Obs,In}$ be a S-family of partial congruences which satisfies the following condition:*

$$\forall obs \in Obs.\forall v, w \in A_{obs}[X_{In}].(v\ R_{A,obs}^{Obs,In}\ w \ \Leftrightarrow \ v = w)$$

*then*

$$\forall s \in S.\forall v, w \in A[X_{In}].v\ R_{A,s}^{Obs,In}\ w \ \Rightarrow \ v \approx_{A,s}^{Obs,In} w$$

**Proof sketch:**
It follows by context induction. The proof of the general case uses that $R$ is a $S$-family of partial congruences which coincides with the set theoretical equality for observable sorts.

The following lemma can also be found in [5]:

**Lemma 4.2** *The sentence $Indist\_rel[(S,Op)[\sim], Obs, In, s]$ for any sort $s \in S$ and for any free variables $x, y \in X_s$ satisfies the following condition which we will refer as the indistinguishability condition:*

$$\forall s \in S.\forall A \in Alg(\Sigma).\forall \rho \in \{x, y\} \ \rightarrow \ A.$$

$$[\![Indist\_rel[\Sigma[\sim], Obs, In](x, y)]\!]_{\rho,A} \Leftrightarrow \rho(x) \approx_{A,s}^{Obs,In} \rho(y)$$

**Theorem 4.3** *For any specification expression $SP$ with signature $\Sigma = (S, Op)$, for any signature morphism $Copy : \Sigma \rightarrow Copy(\Sigma)$ such that $\Sigma \cap Copy(\Sigma) = \emptyset$, for any $S-$family of symbols $\sim$ and $\pi$ the two following equivalences between specification expressions hold:*

$$\textbf{behaviour}\ \ SP\ \ \textbf{wrt}\ \ \approx_{Obs,In}\ \ = \ GBSP[SP, Copy, \Sigma[\sim, \pi_{Copy}], Obs, In]|_\Sigma$$

$$SP/\approx_{Obs,In}\ \ = \ GBSP[SP, Copy, Copy(\Sigma)[\sim, \pi_{Copy^{-1}}], Copy(Obs), Copy(In)]|_\Sigma$$

**Proof:**

- **behaviour** $SP$ **wrt** $\approx_{Obs,In} = GBSP[SP, Copy, \Sigma[\sim, \pi_{Copy}], Obs, In]|_\Sigma$.
  Assume that $BEHSP = BSP[\Sigma[\sim, \pi_{Copy}], Obs, In]$. We differentiate two cases:
  - $A \in Models($**behaviour** $SP$ **wrt** $\approx_{Obs,In}) \Rightarrow$
    $A \in Models(GBSP[SP, Copy, \Sigma[\sim, \pi_{Copy}], Obs, In]|_\Sigma)$.
    Assume that $A \in Models($**behaviour** $SP$ **wrt** $\approx_{Obs,In})$. The proposition which we have to prove is equivalent to

    $$\exists B \in Models(GBSP[SP, Copy, \Sigma[\sim, \pi_{Copy}], Obs, In]).B|_\Sigma = A$$

    To prove this proposition we will build an algebra $B \in Models(GBSP$ $[SP, Copy, \Sigma[\sim, \pi_{Copy}], Obs, In])$ such that $B|_\Sigma = A$. The algebra $B$ is defined as follows:

* $B|_{\Sigma} = A$.
* $B|_{Copy(\Sigma)} = A/{\approx}_{A}^{Obs,In}|_{Copy-1}$.
* $B_{\pi_s} = \epsilon_{A,s}$ for every sort $s \in S$.
  where
  $\epsilon_{A,s}(v) = [v]_{{\approx}_{A,s}^{Obs,In}}$ for all $v \in A_s$.
* $B_{\sim_s} = {\approx}_{A,s}^{In,Obs}$ for every sort $s \in S$

To show that $B \in Models(GBSP[SP, Copy, \Sigma[\sim, \pi_{Copy}], Obs, In])$ we have to show that $B|_{Copy(\Sigma)}|_{Copy} \in Models(SP)$ which obviously holds since $B|_{Copy(\Sigma)}|_{Copy} = A/{\approx}_{A}^{Obs,In}$ and $A/{\approx}_{A}^{Obs,In} \in Models(SP)$.
We have to show also that

$$B \in Models(< \Sigma[\sim, \pi_{Copy}], \\ Indist\_rel[\Sigma[\sim], Obs, In] \cup pEpi[\Sigma[\sim, \pi_{Copy}], Obs, In] >)$$

This proposition holds because $B \models Indist\_rel[\Sigma[\sim], Obs, In]$ and $B \models pEpi[\Sigma[\sim, \pi_{Copy}], Obs, In]$.

$B \models Indist\_rel[\Sigma[\sim], Obs, In]$ holds since for every sort $s$ $B_{\sim_s} = {\approx}_{B|_{\Sigma},s}^{In,Obs}$ and $Indist\_rel[\Sigma[\sim], Obs, In]$ satisfies the indistinguishability condition, and $B \models pEpi[\Sigma[\sim, \pi_{Copy}]]$ holds since for every sort $s$, $B_{\pi_s}$ can be seen as an epimorphism between $A$ and $A/{\approx}_{A,s}^{Obs,In}$ and therefore $B$ satisfies $Hom[\Sigma[\pi_{Copy}, Obs, In], Epihom[\Sigma[\pi_{Copy}], Obs, In]$ and also $\sim -comp[\Sigma[\sim, \pi_{Copy}], Obs, In]$.

– $A \in Models(GBSP[SP, Copy, \Sigma[\sim, \pi_{Copy}], Obs, In]|_{\Sigma}) \Rightarrow$
  $A \in Models(\textbf{behaviour} \quad SP \quad \textbf{wrt} \quad {\approx}_{Obs,In})$.

Assume that $A \in Models(GBSP[SP, Copy, \Sigma[\sim, \pi_{Copy}], Obs, In]|_{\Sigma})$.
This proposition can be rewritten to

$$\exists B \in Models(GBSP[SP, Copy, \Sigma[\sim, \pi_{Copy}], Obs, In]).B|_{\Sigma} = A$$

Let $B$ be a $Signature(GBSP[SP, Copy, \Sigma[\sim, \pi_{Copy}], Obs, In])$-algebra such that $B \in Models(GBSP[SP, Copy, \Sigma[\sim, \pi_{Copy}], Obs, In])$ and $B|_{\Sigma} = A$. By the definition of $Models(GBSP[SP, Copy, \Sigma[\sim, \pi_{Copy}], Obs, In])$ we have that $B \in Models(BEHSP)$ and because of the indistinguishability condition together with the definition of $pEpi[\Sigma[\sim, \pi_{Copy}]]$ we have that

$$\exists C \in Alg(Copy(\Sigma)).C|_{Copy} \cong A/{\approx}_{A}^{Obs,In} \ \& \ B|_{Copy(\Sigma)} = C$$

Let C be a $Copy(\Sigma)$-algebra such that $C|_{Copy} \cong A/{\approx}_{A}^{Obs,In}$ and $B|_{Copy(\Sigma)} = C$. By the definition of $Models(GBSP[SP, Copy, \Sigma[\sim, \pi_{Copy}], Obs, In]|_{\Sigma})$ and since $B|_{Copy(\Sigma)} = C$ we have that $C \in$

$Models(rename \quad SP \quad \textbf{by} \quad Copy)$ and since $C|_{Copy} \cong A/\approx_A^{Obs,In}$ and $SP$ is closed under isomorphism we have that $A/\approx_A^{Obs,In} \in Models(SP)$ and therefore

$$A \in Models(\textbf{behaviour} \quad SP \quad \textbf{wrt} \quad \approx_{Obs,In})$$

- $SP/\approx_{Obs,In} = GBSP[SP, Copy, Copy(\Sigma)[\sim, \pi_{Copy-1}], Copy(Obs), Copy(In)]|_\Sigma$.
  Assume that
  $QGBSP = GBSP[SP, Copy, Copy(\Sigma)[\sim, \pi_{Copy-1}], Copy(Obs), Copy(In)]|_\Sigma$.
  We differentiate two cases:
  - $A \in Models(SP/\approx_{Obs,In}) \Rightarrow A \in Models(QGBSP)$

    Assume that $A \in Models(SP/\approx_{Obs,In})$. The proposition which we have to prove is equivalent to

    $$\exists B \in Models(QGBSP) \; . \; B|_\Sigma = A$$

    To prove this proposition we will build an algebra

    $$B \in Models(QGBSP)$$

    such that $B|_\Sigma = A$. By the proposition $A \in Models(SP/\approx_{Obs,In})$ we know that

    $$\exists A' \in Alg(Signature(SP)).A'/\approx_{A'}^{Obs,In} \cong A \;\&\; A' \in Models(SP)$$

    Let $A'$ be a $Signature(SP)$-algebra such that $A'/\approx_A^{Obs,In} \cong A$ and $A' \in Models(SP)$ and let $h : A'/\approx_{A'}^{Obs,In} \to A$ be the isomorphism which relates each other. The algebra $B$ is defined as follows:
    - $B|_\Sigma = A$.
    - $B|_{Copy(\Sigma)} = A'|_{Copy-1}$.
    - $B_{\pi_{Copy(s)}} = \epsilon_{A'|_{Copy-1}, Copy(s)}$ for every sort $s \in S$.
      where
      $\epsilon_{A,Copy(s)}(v) = h([v]_{\approx_{A,Copy(s)}^{Copy(Obs),Copy(In)}})$
    - $B_{\sim_{Copy(s)}} = \approx_{A'|_{Copy-1},Copy(s)}^{Copy(In),Copy(Obs)}$

    To show that $B \in Models(QGBSP)$ we have to show that $B|_{Copy(\Sigma)}|_{Copy} \in Models(SP)$ which obviously holds since $B|_{Copy(\Sigma)}|_{Copy} = A'$ and we have to show also that

    $$B \in Models(BSP[Copy(\Sigma)[\sim, \pi_{Copy-1}], Copy(Obs), Copy(In)])$$

    which holds since $B \models Indist\_rel[Copy(\Sigma)[\sim], Obs, In]$ and $B \models pEpi[\Sigma[\sim, \pi_{Copy-1}]]$.

$B \models Indist\_rel[Copy(\Sigma)[\sim], Obs, In]$ holds since for every sort $s$ $B_{\sim_{Copy(s)}} = \approx^{Copy(In),Copy(Obs)}_{B|_{Copy(\Sigma)},Copy(s)}$ and $Indist\_rel[Copy(\Sigma)[\sim], Copy(Obs), Copy(In)]$ satisfies the indistinguishability condition. $B \models pEpi[Copy(\Sigma)[\sim, \pi_{Copy^{-1}}]]$ holds since for every sort $s$, $B_{\pi_{Copy(s)}}$ can be seen as an epimorphism between $A'$ and $A$ and therefore $B$ satisfies $Hom[\Sigma[\sim, \pi_{Copy}], Obs, In]$, $Epihom[\Sigma[\pi_{Copy}], Obs, In]$ and also $\sim -comp[\Sigma[\sim, \pi_{Copy}], Obs, In]$.

- $A \in Models(QGBSP) \Rightarrow A \in Models(SP/\approx_{Obs,In})$. Assume that $A \in Models(QGBSP)$. By this proposition we know that there exists a $Signature(QGBSP)$-algebra $B$ such that $B \in Models(QGBSP)$ and $B|_\Sigma = A$. By the definition of $Models(QGBSP)$ we have that $B|_{Copy(\Sigma)}|_{Copy} \in Models(SP)$ and together with the definitions of $Indist\_rel[Copy(\Sigma)[\sim], Copy(Obs), Copy(In)]$ and $pEpi[Copy(\Sigma)[\sim, \pi_{Copy^{-1}}], Copy(Obs), Copy(In)]$ we have that $A \cong B|_{Copy(\Sigma)}|_{Copy}/\approx^{Obs,In}_A$. Therefore, by the definition of $Models(SP/\approx^{Obs,In}_A)$ we have that $A \in Models(SP/\approx^{Obs,In}_A)$.

**Definition 4.30** *The proof system $\Pi^{ASL}_{HOL}$ is inductively defined for the specific operators of the language as follows:*

$$Rules(\Pi^{ASL}_{HOL}, \textbf{reach} \quad SP \quad \textbf{with} \quad (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R})) = Rules(\Pi^{ASL}_{HOL}, SP)$$

$$Symbols(\Pi^{ASL}_{HOL}, \textbf{reach} \quad SP \quad \textbf{with} \quad (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R})) = Symbols(\Pi^{ASL}_{HOL}, SP)$$

$$\Gamma env(\Pi^{ASL}_{HOL}, \textbf{reach} \quad SP \quad \textbf{with} \quad (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R})) = \Gamma env(\Pi^{ASL}_{HOL}, SP) \cup$$
$$Reach\_ax[Signature(SP), (\mathcal{S}_\mathcal{R}, \mathcal{F}_\mathcal{R})]$$

$$Rules(\Pi^{ASL}_{HOL}, \textbf{behaviour} \quad SP \quad \textbf{wrt} \quad \approx_{Obs,In}) =$$
$$Copy''(Rules(\Pi^{ASL}_{HOL}, SP))$$

$$Symbols(\Pi^{ASL}_{HOL}, \textbf{behaviour} \quad SP \quad \textbf{wrt} \quad \approx_{Obs,In}) =$$
$$Copy''(Symbols(\Pi^{ASL}_{HOL}), SP) \cup \Sigma[\sim, \pi_{Copy}]$$

$$\Gamma env(\Pi^{ASL}_{HOL}, \textbf{behaviour} \quad SP \quad \textbf{wrt} \quad \approx_{Obs,In}) =$$
$$\Gamma env(rename \quad SP \quad \textbf{by} \quad Copy) \cup$$
$$\Gamma env(BSP[\Sigma[\sim, \pi_{Copy}], Obs, In])$$

$$Rules(\Pi^{ASL}_{HOL}, \textbf{abstract} \quad SP \quad \textbf{by} \quad \equiv_{Obs,In}) =$$

$$Rules(\Pi^{ASL}_{HOL}, \textbf{behaviour} \quad SP/\approx_{Obs,In} \quad \textbf{wrt} \quad \approx)$$

$$Symbols(\Pi^{ASL}_{HOL}, \textbf{abstract} \quad SP \quad \textbf{by} \quad \equiv_{Obs,In}) =$$

$$Symbols(\Pi^{ASL}_{HOL}, \textbf{behaviour} \quad SP/\approx_{Obs,In} \quad \textbf{wrt} \quad \approx)$$

$$\Gamma env(\Pi^{ASL}_{HOL}, \textbf{abstract} \quad SP \quad \textbf{by} \quad \equiv_{Obs,In}) =$$

$$\Gamma env(\Pi^{ASL}_{HOL}, \textbf{behaviour} \quad SP/\approx_{Obs,In} \quad \textbf{wrt} \quad \approx)$$

$$Rules(\Pi^{ASL}_{HOL}, SP/\approx_{Obs,In}) = Copy''(Rules(\Pi^{ASL}_{HOL}, SP))$$

$$Symbols(\Pi^{ASL}_{HOL}, SP/\approx_{Obs,In}) =$$

$$Copy''(Symbols(\Pi^{ASL}_{HOL}, SP)) \cup Copy(\Sigma)[\sim, \pi_{Copy-1}]$$

$$\Gamma env(\Pi^{ASL}_{HOL}, SP/\approx_{Obs,In}) = \Gamma env(\Pi^{ASL}_{HOL}, rename \quad SP \quad \textbf{by} \quad Copy) \cup$$

$$\Gamma env(BSP[Copy(\Sigma)[\sim, \pi_{Copy-1}], Copy(Obs), Copy(In)])$$

*where $\Sigma$ is the signature of the argument specification $SP$ for every case and the overloaded function symbol $\sigma''$ is used here as the pushout of the following diagram:*

$$\begin{array}{ccc} Sym(\Pi^{ASL}_{HOL}, SP) & \xrightarrow{\sigma''} & PO(i, \sigma'') \\ {\scriptstyle i}\uparrow & & \uparrow \\ \Sigma & \xrightarrow{\quad\sigma\quad} & \Sigma' \end{array}$$

*where $i : \Sigma \hookrightarrow Sym(\Pi^{ASL}_{HOL}, SP)$ and it is also used as the renaming function of environments and of the proof system $\Pi^{ASL}_{HOL}(SP)$ which use the pushout just described , and the overloaded symbol $Copy''$ is the pushout morphism of the following diagram:*

$$\begin{array}{ccc} Sym(\Pi^{ASL}_{HOL}, SP) & \xrightarrow{Copy''} & PO(i, Copy) \\ {\scriptstyle i}\uparrow & & \uparrow \\ \Sigma & \xrightarrow{\quad Copy\quad} & Copy(\Sigma) \end{array}$$

*where* $i : \Sigma \hookrightarrow Symbols(\Pi^{ASL}_{HOL}, SP)$ *and*

$$Copy''(Symbols(\Pi^{ASL}_{HOL}, SP)) \cap \Sigma[\sim, \pi_{Copy}] \; =$$

$$Copy''(\Sigma)$$

*The symbol* $Copy''$ *is also used as the renaming functions of environmnents and of set of rules using the pushout morphism just described with the same name.*

Now we present a proof system of the higher order logic $HOL$ $\Pi_{HOL}(\Gamma)$. This proof system is split in three different kind of rules: derivation rules, typing rules and proof rules which determine the $\beta$-equality. All of them are standard rules presented in a natural deduction style.

**Definition 4.31** *The natural deduction system* $\Pi_{HOL}(\Gamma)$ *is defined by the following set of rules for any* $\Gamma \in \mathcal{P}(|Sen_{AINS}(\Sigma)|)$*:*

$$\frac{\Gamma \cup \phi \Rightarrow_X \phi'}{\Gamma \Rightarrow_X \phi \supset \phi'} \quad (\supset \; i)$$

$$\frac{\Gamma \Rightarrow_X \phi \supset \phi' \quad \Gamma \Rightarrow_X \phi}{\Gamma \Rightarrow_X \phi'} \quad (\supset \; e)$$

$$\frac{\Gamma \Rightarrow_{X \cup x:\tau} \phi}{\Gamma \Rightarrow_X \forall x : \tau.\phi} \quad (\forall \; i)$$

$$\frac{\Gamma \Rightarrow_X \forall x : \tau.\phi \quad X \blacktriangleright t : \tau}{\Gamma \Rightarrow_X \phi\{t/x\}} \quad (\forall \; e)$$

$$\frac{\Gamma \Rightarrow_X \phi \quad \phi \; =_{X,\beta} \; \phi'}{\Gamma \Rightarrow_X \psi} \quad (CONV)$$

*where the set of typing rules of this proof system is:*

$$\frac{}{X \blacktriangleright x_\tau : \tau}\ x_\tau \in X_\tau \qquad (ASS)$$

$$\frac{X \blacktriangleright t_1 : s_1 \quad \ldots \quad X \blacktriangleright t_n : s_n}{X \blacktriangleright f(t_1, \ldots, t_n) : s}\ f : s_1 \times \ldots \times s_n \to s \in \Sigma \qquad (APPL)$$

$$\frac{X \cup \{x_1 : \tau_1, \ldots, x_n : \tau_n\} \blacktriangleright \phi : \mathbf{Prop}}{X \blacktriangleright \lambda(x_1 : \tau_1, \ldots, x_n : \tau_n).\phi : [\tau_1, \ldots, \tau_n]} \qquad (\lambda ABS)$$

$$\frac{X \blacktriangleright t_1 : \tau_1 \quad \ldots \quad X \blacktriangleright t_n : \tau_n \quad X \blacktriangleright t : [\tau_1, \ldots, \tau_n]}{X \blacktriangleright t(t_1, \ldots, t_n) : \mathbf{Prop}} \qquad (\lambda APPL)$$

$$\frac{X \cup x : \tau \blacktriangleright \phi : \mathbf{Prop}}{X \blacktriangleright \forall x : \tau.\phi : \mathbf{Prop}} \qquad (\forall)$$

$$\frac{X \blacktriangleright \phi : \mathbf{Prop} \quad X \blacktriangleright \phi' : \mathbf{Prop}}{X \blacktriangleright \phi \supset \phi' : \mathbf{Prop}} \qquad (\supset)$$

*and the $\beta$-equality rules are:*

$$\frac{}{x_\tau =_{\beta,X} x_\tau}\ x \in X_\tau \qquad (Vareq)$$

$$\frac{\begin{array}{c} X \blacktriangleright t_1 : s_1 \quad \ldots \quad X \blacktriangleright t_n : s_n \\ t_1 =_{\beta,X} t'_1 \ldots t_n =_{\beta,X} t'_n \end{array}}{f(t_1, \ldots, t_n) =_{\beta,X} f(t'_1, \ldots, t'_n)}\ f : s_1 \times \ldots \times s_n \to s_n \in \Sigma \qquad (Termeq)$$

$$\frac{\begin{array}{c} X \blacktriangleright t\ (t_1, \ldots, t_n) : \mathbf{Prop} \quad t\ =_{\beta,X}\ t' \\ t_1\ =_{\beta,X}\ t'_1 \ldots t_n =_{\beta,X} t'_n \end{array}}{t(t_1, \ldots, t_n)\ =_{\beta,X}\ t'(t'_1, \ldots, t'_n)} \qquad (Appleq)$$

$$\frac{\begin{array}{c} X \blacktriangleright t_1 : \tau_1 \quad \ldots \quad X \blacktriangleright t_n : \tau_n \\ X \blacktriangleright \lambda(x_1 : \tau_1, \ldots, x_n : \tau_n).\phi : [\tau_1, \ldots, \tau_n] \\ \phi\ \{t_1/x_1\} \ldots \{t_n/x_n\}\ =_{\beta,X}\ \phi' \end{array}}{\lambda(x_1 : \tau_1, \ldots, x_n : \tau_n).\phi\ (t_1, \ldots, t_n)\ =_{\beta,X}\ \phi'} \qquad (Llambdaeq)$$

$$\frac{\begin{array}{c} X \blacktriangleright \lambda(x_1:\tau_1,\ldots,x_n:\tau_n).\phi : [\tau_1,\ldots,\tau_n] \\ \phi \ =_{\beta,X\cup\{x_1:\tau_1,\ldots,x_n:\tau_n\}} \ \phi'\ \{x_1/x_1'\}\ldots\{x_n/x_n'\} \end{array}}{\lambda(x_1:\tau_1,\ldots,x_n:\tau_n).\phi \ =_{\beta,X} \ \lambda(x_1':\tau_1,\ldots,x_n':\tau_n).\phi'} \ (Lambdaeq)$$

$$\frac{\begin{array}{c} X\cup\{x:\tau\} \blacktriangleright \phi : \mathbf{Prop} \\ \phi \ =_{\beta,X\cup\{x:\tau\}} \ \phi'\{x/x'\} \end{array}}{\forall x:\tau.\phi \ =_{\beta,X} \ \forall x':\tau.\phi'} \ (Foralleq)$$

$$\frac{\begin{array}{c} X \blacktriangleright \phi' : \mathbf{Prop} \\ X \blacktriangleright \phi : \mathbf{Prop} \\ \phi' \ =_{\beta,X} \ \phi \end{array}}{\phi \ =_{\beta,X} \ \phi'} \ (Sym)$$

**Definition 4.32** *The encoding of the logical operators* **false**, **true**, $=, \vee, \wedge, \exists$ *is as follows:*

$$\begin{array}{l} false =_{def} \forall P : \mathbf{Prop}.P \\ true =_{def} \forall P : \mathbf{Prop}.P \supset P \\ t =_\tau r =_{def} \forall P : [\tau].P\ t \supset P\ r \\ \phi \wedge \phi' =_{def} \forall P : \mathbf{Prop}.(\phi \supset \phi' \supset P) \supset P \\ \phi \vee \phi' =_{def} \forall P : \mathbf{Prop}.(\phi \supset P) \supset (\phi' \supset P) \supset P \\ \exists x:\tau.\phi =_{def} \forall P : \mathbf{Prop}.((\forall x:\tau.\phi) \supset P) \supset P \end{array}$$

**Proposition 4.2** *The following set of rules is admissible in the proof system* $\Pi_{HOL}(\Gamma)$ *for any environment* $\Gamma \in \mathcal{P}(|Sen_{AINS}(\Sigma)|)$*:*

$$\frac{}{\{\} \Rightarrow_X \mathbf{true}} \ (T) \qquad \frac{}{\Gamma \Rightarrow_X \mathbf{false} \supset \phi} \ (F)$$

$$\frac{\Gamma \Rightarrow_X \phi_1 \wedge \phi_2}{\Gamma \Rightarrow_X \phi_1} \ (\wedge El) \qquad \frac{\Gamma \Rightarrow_X \phi_1 \wedge \phi_2}{\Gamma \Rightarrow_X \phi_2} \ (\wedge Er)$$

$$\frac{\Gamma \Rightarrow_X \phi_1 \quad \Gamma \Rightarrow_X \phi_2}{\Gamma \Rightarrow_X \phi_1 \wedge \phi_2} \ (\wedge I)$$

$$\frac{\Gamma \Rightarrow_X \phi_1}{\Gamma \Rightarrow_X \phi_1 \vee \phi_2} \ (\vee Il) \qquad \frac{\Gamma \Rightarrow_X \phi_2}{\Gamma \Rightarrow_X \phi_1 \vee \phi_2} \ (\vee Ir)$$

$$\frac{\Gamma \Rightarrow_X \phi_1 \vee \phi_2 \quad \Gamma \Rightarrow_X \phi_1 \supset \psi \quad \Gamma \Rightarrow_X \phi_2 \supset \psi}{\Gamma \Rightarrow_X \psi} \ (\vee E)$$

$$\frac{X \ \blacktriangleright \ t:\tau \quad \Gamma \cup \{t:\tau\} \Rightarrow_X \phi\{t/x\}}{\Gamma \Rightarrow_X \exists x:\tau.\phi} \ (\exists I)$$

$$\frac{\Gamma \Rightarrow_X \exists x : \tau.\phi \quad \Gamma \cup \{\phi\} \Rightarrow_{X \cup \{x:\tau\}} \psi}{\Gamma \Rightarrow_X \psi} \quad (\exists E)$$

$$\frac{\Gamma \cup \{\phi\} \Rightarrow_X \mathbf{false}}{\Gamma \Rightarrow_X \neg\phi} \quad (\neg I) \qquad \frac{\Gamma \Rightarrow_X \psi \quad \Gamma \Rightarrow_X \neg\psi}{\Gamma \Rightarrow_X \phi} \quad (\neg E)$$

And next, we present the proof of soundness and incompleteness of $\vdash_{\Pi^{ASL}_{HOL}}$.

**Theorem 4.4** *The consequence relation* $\vdash_{\Pi^{ASL}_{HOL}}$ *is sound.*

**Proof:**
The proof is by induction of specification expression in a similar way as the proof of soundness of $\vdash_{\Pi^{ASL}_{FOLEQ}}$ where the proofs for the cases of the behaviour and quotient operator, theorem 2.42 is used.

**Theorem 4.5** *There is no sound and complete consequence relation of the form* $\vdash_{\Pi^{ASL}_{HOL}}$ *for ASL.*

**Proof:**
It follows in the same way as the incompleteness of $\Pi^{ASL}_{FOLEQ}$

# 5 Conclusions

In this paper, we have presented finitary versions of a non-compositional and infinitary proof system for the deduction of sentences from $ASL$ specification in first-order and higher-order logic. These proof systems are inductively defined by specification expressions and the most interesting parts of the new design decisions are in the reachability and behavioural operators. For first-order logic, since the logic is not very expressive, it is needed to add proof rules to define the proof systems of the reachability operator and proof rules to define the observational equality associated to the behavioural operators together with proof rules to reason about context induction. For higher-order logic, since the logic is quite expressive, we can add axioms to define the proof system for the reachability operator and axioms to define the observational equality based on the idea that this equality is the greatest congruence which coincide with the set-theoretical equality in observable sorts.

The main application of this finitary proof systems is to give adequate representations of the proof systems in type-theoretic logical frameworks.

# References

[1] Philippa Gardner. *Representing Logics in Type Theory.* PhD thesis, University of Edinburgh, July 1992.

[2] Joseph A Goguen and Rod Burstall. INSTITUTIONS: Abstract model theory for specification and programming. *Journal of the Assoc. for Computing Machinery*, 39(1):95–146, 1992.

[3] Rolf Hennicker. *Structured Specifications with Behavioural Operators: Semantics, Proof Methods and Applications.* Habilitationsschrift, Institut für Informatik, Ludwig-Maximilians-Universität München, June 1997.

[4] Rolf Hennicker, Martin Wirsing, and Michel Bidoit. Proof systems for structured specifications with observability operators. *Theoretical Computer Science*, 173, February 1997.

[5] Martin Hofmann and Donald Sannella. On behavioural abstraction and behavioural satisfaction in higher-order logic. *Theoretical Computer Science*, 167:3–45, 1996.

[6] D. MacQueen and D. Sannella. Completeness of proof systems for equational specifications. *IEEE Transactions on Software Engineering*, SE-11(454–461), 1985.

[7] Nikos Mylonakis. *A type-theoretic approach to proof support for algebraic design frameworks.* PhD thesis, Universitat Politècnica de Catalunya, 2000. To appear.